

# ESPECIFICAÇÃO E PROJETO DE UM ROBÔ ATUADO POR CABOS

**MATHEUS CAMPOS OLIVIERI**

setembro de 2017

**Instituto Superior de Engenharia do Porto**  
Departamento de Engenharia Eletrotécnica  
Rua Dr. António Bernardino de Almeida 431, 4200-072 Porto

# **ESPECIFICAÇÃO E PROJETO DE UM ROBÔ ATUADO POR CABOS**

Tese/Dissertação do Mestrado em Engenharia Eletrotécnica e de Computadores -  
Área de Especialização em Sistema de Planeamento Industrial

**Matheus Campos Olivieri, 1160389@isep.ipp.pt**

Orientação científica: Prof. Manuel Fernando dos Santos Silva, mss@isep.ipp.pt

Coorientação: Prof. Pedro Costa, pedrogc@fe.up.pt;

Prof. Roberto Ribeiro Neli, neli@utfpr.edu.br

Ano Lectivo: 2016-2017





---

# Agradecimentos

---

As minhas primeiras palavras de agradecimento devem ir diretamente a minha família, sem ela nada disto seria possível. A minha mãe, Marilda, por sempre me dizer as palavras certas na hora certa, me dando amor e afeto, permitindo meu equilíbrio emocional durante toda esta trajetória. Ao meu pai, José Carlos, por não medir esforços para me formar, isto no âmbito académico e social, se cheguei ao fim desta etapa foi graças ao seu apoio e a maneira como me ensinou a ver o mundo.

Um agradecimento especial ao meu orientador científico, Prof. Manuel Fernando dos Santos Silva, por toda a paciência e atenção que tiveste comigo, desde o momento da escolha do tema até as últimas revisões. Agradeço o esforço em fazer com que este trabalho acontecesse da melhor forma. Aproveito para agradecer aos meus coorientadores científicos, Prof. Pedro Costa e Prof. Roberto Ribeiro Neli, por suas propostas, críticas e opiniões buscando sempre um melhor resultado. Agradeço ao Prof. Germano Veiga por todo seu aconselhamento no ponto de vista mecânico deste projeto.

Não posso deixar de agradecer às instituições que participaram da minha formação: a Universidade Tecnológica Federal do Paraná por me encorajar a atravessar o oceano e desenvolver este trabalho em terras portuguesas junto ao Instituto Superior de Engenharia do Porto. Agradeço de forma geral a todos os professores e colaboradores das duas instituições. Não poderia deixar de agradecer ao INESC TEC por disponibilizar os recursos necessários para que este trabalho fosse desenvolvido.

Por último, mas não menos especial, um muito obrigado aos meus amigos, em especial aqueles que estiveram comigo durante esta fase, sem eles esta etapa certamente seria muito mais difícil.

Obrigado!



---

# Resumo

---

A evolução da humanidade é fator responsável pelo surgimento de necessidades, que hoje são consideradas básicas, e que antes não existiam. A moradia, por exemplo, tornou-se uma necessidade básica do ser humano, e passou a possuir um alto custo envolvido. O aumento populacional provocou a intensificação da procura por moradias, o que acabou tornando a mão de obra cada vez mais cara. Para tentar solucionar este problema várias soluções foram apresentadas, dentre elas tem-se a construção autónoma das moradias.

Este trabalho apresenta uma proposta de um robô paralelo atuado a cabos com a finalidade de ser utilizado na construção civil, levando em consideração a carga útil e o tamanho do espaço de trabalho para propor a arquitetura.

Para chegar a uma proposta condizente, é feito um estudo sobre os robôs já existentes e quais os tipos de atuadores e sensores utilizados nos robôs, a partir disso é proposto uma arquitetura para o sistema.

O sistema proposto consiste em quatro pilares com distâncias variáveis entre si, sendo necessário apenas informá-las no sistema de controlo. No topo de cada um dos pilares há uma roldana multi-direcional por onde passa o cabo que sai da base de cada pilar e se conecta a uma plataforma no interior do volume de trabalho. A plataforma é o único ponto do robô onde se conectam os quatro cabos, sendo conectada a ela o atuador final, que neste caso é uma garra que é responsável pela tarefa de *pick-and-place*.

A partir do sistema descrito é implementado um *software* para a simulação do sistema, o qual apresenta em uma janela gráfica o movimento planeado. Posteriormente este *software* é colocado a prova e analisados os resultados obtidos.

Esta dissertação apresenta o processo de planeamento de um robô atuado por quatro cabos, desde o estudo da cinemática, a arquitetura e a escolha dos dispositivos necessários para a implementação.

## Palavras chave

Robô a cabos, Cinemática, Planeamento da trajetória, Controlo, Simulação.



---

# Abstract

---

With the evolution of humanity, some basic necessities appeared that would become very expensive in the future, one of them is the housing, with the population increase the demand for housing intensified, and the labor became more and more expensive. In order to solve this problem several solutions were presented, among them the autonomous construction of the dwellings.

This work presents a proposal for a cable-driven parallel robot, it aims to be used in civil construction, considering the useful load and the workspace size to propose the architecture.

In order to achieve a suitable proposal, a study about the existing robots and the types of actuators and sensors used in the robots is made, concluding an architecture for the system.

The proposed system consists in four pillars with variable distances between them, being only necessary to inform that distances in the control system. At the top of each pillar there is a multi-directional pulley where the cable from the base of the pillar passes and connects to a platform inside the working volume. This platform is the only point where the four cables are connected in the robot. It also connects the final actuator, in this case is a gripper which is responsible for the pick-and-place task.

From the described system a software is implemented for its simulation, which presents in a graphic window the planned movement. Subsequently this software is tested and the results obtained are analyzed.

This thesis presents the process of planning a robot operated by four cables, from the study of the kinematics, architecture and the choice of the necessary devices for the implementation.

## Keywords

Cable-driven parallel robots, Kinematics, Trajectory planning, Control, Simulation.



---

# Conteúdo

---

Conteúdo	ix
Lista de Figuras	xiii
Lista de Tabelas	xv
Acrónimos	xvii
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Calendarização . . . . .	2
1.4 Estrutura do Relatório . . . . .	3
<b>2 Estado da Arte</b>	<b>5</b>
2.1 Robótica Industrial . . . . .	5
2.2 Robôs Atuados por Cabos . . . . .	6
2.2.1 <i>The NIST RoboCrane</i> . . . . .	7
2.2.2 <i>Cable-Driven Parallel Robots IPAnema</i> . . . . .	9
2.2.3 <i>REELAX: Configurable Robots of Small Size</i> . . . . .	11
2.2.4 <i>CABLEBOT – Maintance and Logistics of Large Scale Pro-</i> <i>ducts</i> . . . . .	15
2.3 Conclusão do Capítulo . . . . .	15
<b>3 Atuadores e Sensores para Robôs a Cabos</b>	<b>17</b>
3.1 Atuadores . . . . .	17
3.1.1 Guinchos . . . . .	17
3.1.2 Motores de Passo . . . . .	18
3.1.3 Servomotor com Redutor e Codificador . . . . .	20
3.2 Sensores . . . . .	20
3.2.1 Codificador . . . . .	21



3.2.2	Sensor de Força . . . . .	22
3.3	Conclusão do Capítulo . . . . .	23
<b>4</b>	<b>Cinemática e Planeamento das Trajetórias dos Robôs Atuados por Cabos</b>	<b>25</b>
4.1	Modelo no Plano $x - y$ . . . . .	25
4.2	Modelo em 3 Dimensões (3D) . . . . .	28
4.2.1	Cinemática Inversa . . . . .	29
4.2.2	Cinemática Direta . . . . .	29
4.3	Orientação . . . . .	30
4.3.1	Orientação da Plataforma . . . . .	30
4.3.2	Orientação da Garra . . . . .	33
4.4	Planeamento das Trajetórias . . . . .	34
4.4.1	Planeamento da Trajetória Linear no Espaço das Juntas . .	34
4.4.2	Planeamento da Trajetória Linear no Espaço Operacional .	35
4.5	Conclusão do Capítulo . . . . .	38
<b>5</b>	<b>Arquitetura do Sistema</b>	<b>41</b>
5.1	Requisitos do Sistema Robótico . . . . .	41
5.2	Arquitetura do Sistema . . . . .	42
5.2.1	Estrutura . . . . .	42
5.2.2	Controlo . . . . .	44
5.2.3	Comunicação . . . . .	45
5.3	Especificação dos Equipamentos Utilizados na Implementação . . .	45
5.3.1	Cabos . . . . .	46
5.3.2	Guincho . . . . .	46
5.3.3	Sistema de Sustentação Superior do Cabo . . . . .	48
5.3.4	Controlo . . . . .	49
5.4	Dispositivos Escolhidos para a Implementação do Sistema . . . . .	49
5.5	Conclusão do Capítulo . . . . .	50
<b>6</b>	<b>Simulação de um Robô Paralelo Atuado a Cabos</b>	<b>51</b>
6.1	<i>Software</i> de Simulação do Sistema . . . . .	51
6.2	Conclusão do Capítulo . . . . .	54
<b>7</b>	<b>Testes e Resultados da Avaliação do Sistema</b>	<b>55</b>
7.1	Simulação do Movimento . . . . .	55
7.2	Erro . . . . .	56
7.3	Conclusão do Capítulo . . . . .	57
<b>8</b>	<b>Conclusão e Desenvolvimentos Futuros</b>	<b>63</b>
8.1	Balanço da Obra . . . . .	63
8.2	Conclusão . . . . .	64

8.3	Desenvolvimentos Futuros . . . . .	64
<b>Bibliografia</b>		<b>65</b>
<b>Apêndice A Parâmetros do Cabo de Aço</b>		<b>69</b>
<b>Apêndice B Dispositivos para Implementação</b>		<b>71</b>
B.1	Opção 1 . . . . .	71
B.2	Opção 2 . . . . .	72
B.3	Opção 3 . . . . .	73
B.4	Opção 4 . . . . .	74
<b>Apêndice C Códigos em Java<sup>®</sup> para Geração do <i>Software</i> de Simulação do Sistema</b>		<b>75</b>
C.1	Código do Cálculo da Cinemática e Planeamento das Trajetórias .	75
C.2	Código da Classe que Realiza o Desenho da Janela de Simulação do Sistema . . . . .	79
C.3	Código da Classe de Comunicação com o Matlab <sup>®</sup> . . . . .	84



---

# Lista de Figuras

---

2.1	Estrutura básica do RoboCrane [15] . . . . .	8
2.2	Sistema de controlo do The NIST RoboCrane [15] . . . . .	8
2.3	Configuração submarina (esquerda) [23] e expedição lunar (direita) [19] . . . . .	9
2.4	IPAnema 1 (esquerda) e IPAnema 2 (direita) [11] . . . . .	10
2.5	Estrutura do sistema de controlo para o robô a cabos IPAnema [11] . . . . .	11
2.6	Vista lateral do ReelAx [26] . . . . .	12
2.7	Vista lateral do ReelAx utilizando quatro pilares [25] . . . . .	13
2.8	Servomotor com tambor para enrolamento acoplado [26] . . . . .	13
2.9	Vista frontal do HPRcable [25] . . . . .	14
2.10	Vista frontal do sistema robótico COGIRO [27] . . . . .	14
2.11	Reparos em aeronave (esquerda) e manuseio de vigas (direita) [28] . . . . .	15
3.1	Esquema mecânico do guincho utilizado na família de robôs IPAnema [11] . . . . .	18
3.2	Vista em corte longitudinal de um motor de passo [29] . . . . .	19
3.3	Sensor de mola com sensor de deslocamento [33] . . . . .	22
4.1	Modelo simplificado do sistema no plano $x - y$ . . . . .	26
4.2	Modelo real do sistema no plano $x - y$ . . . . .	27
4.3	Modelo do sistema real em 3D com plataforma . . . . .	28
4.4	Representação gráfica do sistema utilizado para a aproximação da cinemática direta no plano $x - y$ . . . . .	30
4.5	<i>Yaw, Pitch e Roll</i> na plataforma com referencial cartesiano . . . . .	31
4.6	Sistema com diferentes orientações da plataforma e comprimentos de cabo para o mesmo ponto $C$ . . . . .	31
4.7	Rotação <i>Pitch</i> no sistema . . . . .	32
4.8	Rotação <i>Yaw</i> no sistema . . . . .	33
4.9	Evolução do comprimento (esquerda), velocidade (centro) e aceleração (direita) ao longo do tempo para os cabos $L_1$ (primeira fileira), $L_2$ (segunda fileira), $L_3$ (terceira fileira) e $L_4$ (última fileira) . . . . .	36

4.10	Evolução da posição (esquerda), velocidade (centro) e aceleração (direita) do atuador final em $x$ (primeira fileira), $y$ (segunda fileira) e $z$ (última fileira) ao longo do tempo . . . . .	38
5.1	Modelo 3D da estrutura mecânica . . . . .	42
5.2	Atuador final conectado ao sistema . . . . .	43
5.3	Representação das interações do sistema . . . . .	44
5.4	Vista lateral do sistema robótico completo . . . . .	45
5.5	Vista lateral do esquema mecânico do guincho completo . . . . .	47
5.6	Componentes do sistema de sustentação superior do cabo . . . . .	48
6.1	Janela de interface gráfica . . . . .	52
6.2	Janela de simulação do movimento linear no espaço operacional com a garra aberta . . . . .	53
6.3	Janela de simulação do movimento linear no espaço das juntas com a garra fechada . . . . .	54
7.1	Sequência do movimento nos planos $x - y$ (esquerda) e $x - z$ (direita) com o planeamento da trajetória linear no espaço operacional . . . . .	58
7.2	Sequência do movimento nos planos $x - y$ (esquerda) e $x - z$ (direita) com o planeamento da trajetória linear no espaço das juntas . . . . .	59
7.3	Aspeto geral da simulação utilizando a plataforma com dimensões $0,5 \times 0,5 \times 0,3$ m . . . . .	60
7.4	Aspeto geral da simulação utilizando a plataforma com dimensões $0,1 \times 0,1 \times 0,1$ m . . . . .	61
7.5	Aspeto geral da simulação utilizando apenas um ponto para conexão dos cabos. . . . .	62

---

# Lista de Tabelas

---

1.1	Calendarização do projeto . . . . .	3
3.1	Motor de passo <i>vs</i> servomotor . . . . .	21
5.1	Características do servo motor Dynamixel MX-28T [36] . . . . .	50
7.1	Coordenadas retiradas das simulações . . . . .	56
7.2	Erro absoluto . . . . .	57
A.1	Cargas de rutura e massa linear [35] . . . . .	69
B.1	Primeira solução encontrada . . . . .	71
B.2	Segunda solução encontrada . . . . .	72
B.3	Terceira solução encontrada . . . . .	73
B.4	Quarta solução encontrada . . . . .	74



---

# Acrónimos

---

Abreviatura	Descrição
3D	3 Dimensões
CA	Corrente Alternada
CC	Corrente Contínua
IP	<i>Internet Protocol</i>
MAC	<i>Media Access Control</i>
MRV	Máquina de Relutância Variável
NC	<i>Numeric Controller</i>
NIST	<i>National Institute of Standards and Technology</i>
PLC	<i>Programmable Logic Controller</i>
RIA	<i>Robotic Industries Association</i>
RUR	<i>Rossum's Universal Robots</i>
SCARA	<i>Selective Compliant Assembly Robot Arm</i>
TCP	<i>Tool Center Point</i>
USB	<i>Universal Serial Bus</i>





## Capítulo 1

---

# Introdução

---

### 1.1 Contextualização

Com a evolução da humanidade, o homem sempre procurou construir ferramentas que tornassem o serviço do dia-a-dia mais fácil, ou até mesmo que fizessem o serviço sozinho; isto vem desde a primeira lança até às máquinas mais modernas existentes nas fábricas hoje em dia.

Com a expansão da população mundial, a necessidade de moradias se intensificou, havendo um crescimento significativo no ramo da construção civil. Com uma grande procura, a mão de obra especializada se tornou cada vez mais escassa, resultando em um serviço com um custo mais elevado. Algumas tecnologias foram desenvolvidas para que haja redução de custos deste serviço, sendo uma destas o *Contour Crafting*.

Desenvolvido pelo Dr. Behrokh Khoshnevis na Universidade do Sul da Califórnia, o *Contour Crafting* é uma tecnologia de construção feita em camadas. O sucesso desta tecnologia se dá pelo bom acabamento superficial e pela rapidez na fabricação de peças de grande escala. Estes fatores só se tornaram possíveis, pois o processo fabril é feito por máquinas automatizadas que manuseiam ferramentas anteriormente utilizadas pelo homem [1] [2].

Com o uso de robôs na construção civil, é possível solucionar alguns dos problemas citados acima. A construção se dá em um tempo reduzido, o custo da obra também é reduzido, levando em conta que não é mais necessário o uso de mão de obra especializada na construção.

Os robôs paralelos atuados por cabos têm muito a contribuir neste setor. Os robôs a cabos fazem parte da classe de robôs paralelos, porém seus elos são substituídos por cabos; com isto, sua massa e sua inércia são reduzidos conside-

ravelmente, trazendo inúmeras vantagens, dentre elas a diminuição do risco de acidentes letais com o utilizador. Por se tratar de um sistema leve, e de baixo custo, é possível alcançar grandes volumes de trabalho o que o torna um robô adequado para a construção civil.

## 1.2 Objetivos

O objetivo principal deste trabalho é especificar, projetar e simular um sistema robótico do tipo “*cable robot*”, com um volume de trabalho variável, podendo chegar a até  $75 \text{ m}^3$ , com uma estrutura de  $5 \text{ m} \times 5 \text{ m} \times 3 \text{ m}$  e uma capacidade de carga máxima de 5 kg. Pretende-se que este sistema seja capaz de efetuar construções baseadas em tijolos a partir de uma entrada do utilizador.

O sistema é baseado em uma plataforma conectada a cabos de aço, que por sua vez estão conectados a guinchos elétricos. Conforme os atuadores modificam a tensão dos cabos a plataforma se move no interior do volume de trabalho. Na parte inferior da flange há uma garra, a qual é a responsável por pegar os tijolos e depositá-los no local desejado.

Pretende-se que o volume de trabalho do sistema robótico seja variável, ou seja, a sua estrutura não tem dimensões fixas, podendo ser configurada da forma que melhor atenda a utilização, desde que não ultrapasse os limites impostos.

## 1.3 Calendarização

De modo a serem concretizados os objetivos propostos seguiu-se uma sequência de tarefas durante sete meses de realização deste trabalho, conforme a calendarização vista na Tabela 1.1. Os meses de fevereiro e março foram dedicados a pesquisas sobre o funcionamento dos robôs a cabos, quais os equipamentos similares existentes no mercado; foram estipulados quais requisitos básicos o robô proposto deveria atender.

No mês de abril foram abordados dois temas, na primeira quinzena foi feito o estudo de quais os atuadores e sensores são utilizados nos robôs atuados a cabos, bem como seu funcionamento, e suas características físicas; na segunda quinzena foi proposta uma arquitetura para o sistema robótico.

Durante o mês de maio foi feito o estudo da cinemática para um robô atuado por quatro cabos, e após isto foi feito o planeamento das trajetórias. Durante os meses de junho e julho foi implementado o simulador para o robô, e definidos quais os dispositivos seriam utilizados para a montagem do sistema robótico. O mês de agosto foi utilizado para fazer a revisão da dissertação.

Tabela 1.1: Calendarização do projeto

	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Redação da Dissertação	X	X	X	X	X	X	X
Estudo dos robôs atuados por cabos	X	X					
Estudo dos sensores e atuadores			X				
Desenvolvimento da arquitetura			X				
Estudo da cinemática				X			
Desenvolvimento do planeamento da trajetória				X			
Implementação do <i>software</i> de simulação e controlo					X	X	
Escolha dos dispositivos para implementação					X	X	
Revisão							X

## 1.4 Estrutura do Relatório

Para que a compreensão ocorra de maneira mais suave, este trabalho foi dividido em oito capítulos: introdução, estado da arte, atuadores e sensores para robôs a cabos, cinemática e planeamento das trajetórias dos robôs atuados por cabos, arquitetura do sistema, simulação de um robô atuado por cabos, testes e resultados da avaliação do sistema, e conclusões.

O primeiro capítulo busca fazer uma sucinta introdução ao tema, apresentando o que será tratado neste trabalho, e de qual forma será abordado. Apresenta também qual a motivação e quais os objetivos do mesmo, e por fim introduz a calendarização e a organização do trabalho.

O segundo capítulo contempla o estado da arte e efetua uma pequena introdução histórica sobre a robótica, até chegar à robótica industrial, bem como algumas das principais obras literárias associadas ao assunto. Os robôs atuados a cabos são tratados a seguir, onde há exemplos de sua utilização em diversas áreas de atuação, e alguns grupos que pesquisam o tema.

O Capítulo três apresenta os principais tipos de motores utilizados nos guinchos atuadores do sistema e os sensores utilizados por este tipo de robôs.

O Capítulo quatro esboça a cinemática do sistema, bem como suas equações matemáticas desde o modelo simplificado em duas dimensões até o modelo real em três dimensões; apresenta as orientações relativas e suas equações matemáticas, e por fim apresenta o planeamento das trajetórias.

Os requisitos, especificações do projeto e a arquitetura do robô desenvolvido neste trabalho são apresentados no Capítulo cinco, bem como os componentes adequados para a implementação do sistema.

No Capítulo seis é apresentada a implementação do *software* de simulação para um robô atuado por quatro cabos. O Capítulo sete contempla os testes e resultados do sistema proposto.

O Capítulo oito apresenta a conclusão do trabalho, bem como o balanço entre o que foi proposto, e o que foi alcançado apresentando propostas para desenvolvimentos futuros.

## Capítulo 2

---

# Estado da Arte

---

*Este capítulo é destinado à introdução à robótica, quais os principais grupos que se dedicaram ao estudo e desenvolvimento dos robôs atuados a cabos, e quais as suas principais aplicações na indústria da atualidade.*

### 2.1 Robótica Industrial

Desde os primórdios das sociedades, o homem vem produzindo ferramentas para facilitar a sua vida em algumas atividades. No século XVIII, a revolução industrial estimulou o consumismo das pessoas, e com isso as indústrias começaram a produzir muito mais produtos para suprir a necessidade criada, necessitando de muito mais trabalhadores. No intuito de minimizar o uso da mão de obra, foram surgindo algumas máquinas automatizadas, e com o avanço da eletrônica, estas máquinas tornaram-se cada vez mais complexas e autônomas [3].

O termo *robot* foi utilizado pela primeira vez em uma peça de teatro denominada Rossum's Universal Robots (RUR), do escritor Karel Capek, e fazia referência à palavra checa *robota*, que significa trabalho forçado ou trabalho escravo [4]. O enredo da peça trata de seres biológicos criados em laboratório, com capacidade de pensar por si mesmos. No início eles parecem felizes por trabalhar para os seres humanos, mas uma rebelião dos robôs leva à extinção da raça humana [5].

Anos mais tarde, em 1941, o escritor e cientista Isaac Asimov, utilizou pela primeira vez a palavra robótica, e propôs três leis da robótica, às quais anos depois acrescentou a lei zero [6].

- Lei Zero: um robô não pode fazer mal à humanidade, ou, por inação, permitir que ela sofra algum mal.

- Lei Um: um robô não pode fazer mal a um ser humano e nem, por inação, permitir que algum mal lhe aconteça.
- Lei Dois: um robô deve obedecer a ordens dadas a ele por seres humanos, exceto quando tais ordens entrem em conflito com uma lei de ordem superior.
- Lei Três: um robô deve proteger a sua própria existência, desde que tal proteção não entre em conflito com uma lei de ordem superior.

Em 1962 o primeiro robô foi utilizado em um processo de fabrico. Chamado Unimate, o robô criado pelo inventor americano George Devol, com o apoio de Joseph Engelberger, foi introduzido na fábrica da General Motors, em Nova Jersey, e desempenhava a função de carregar e descarregar a máquina de fundição. Com o Unimate a GM revolucionou o cenário da produção automotiva, chegando a produzir até 110 carros por hora. Não demorou muito para as empresas automotivas europeias, como BMW, Volvo e Mercedes Benz seguirem o exemplo da americana [7]. A segunda geração de robôs se deu início em 27 de julho de 1982, quando o inventor japonês Hiroshi Makino patenteou o braço mecânico *Selective Compliant Assembly Robot Arm* (SCARA) [8]. Este robô é muito utilizado para aplicações de montagem, por exemplo, o preenchimento de placas de circuito impresso, pois diferente dos braços mecânicos série, os robôs SCARA possuem uma complacência seletiva no plano  $x - y$  [3] [9].

Segundo a *Robotic Industries Association* (RIA) para que um dispositivo seja considerado um robô, ele deve ser um manipulador multifuncional reprogramável, projetado para mover materiais, peças, ferramentas ou dispositivos especializados através de movimentos programados variáveis para a realização de uma variedade de tarefas [10].

## 2.2 Robôs Atuados por Cabos

Desde a primeira utilização de um robô em um processo produtivo que têm sido pesquisadas melhorias para os sistemas robóticos em série, e, após anos de trabalho chegou-se à conclusão que o potencial de melhoria de desempenho está a ponto de se esgotar. Por outro lado, os robôs paralelos atuados por cabos podem melhorar critérios críticos para o desempenho de um sistema: volume de trabalho e carga útil são dois deles [11].

A origem dos robôs paralelos atuados por cabos deu-se no final da década de 1980 nos Estados Unidos e no Japão, onde os primeiros conceitos de guindastes elétricos e manipuladores cinemáticos foram combinados com o conceito de dispositivos robóticos paralelos atuado a cabos. Fascinados pelas vantagens apresentadas pelo novo modelo robótico, pesquisadores de todo o mundo voltaram seu olhares para os robôs a cabos [12].

Os robôs a cabos são sistemas capazes de movimentar uma plataforma no interior do seu volume de trabalho e executar funções com uma ferramenta conectada a esta plataforma. O sistema robótico é baseado em guinchos elétricos, que quando liberam ou recolhem os cabos conectados à plataforma variam a tensão e provocam o movimento. A plataforma é a única parte do robô que tem o contacto com todos os cabos que constituem o sistema.

Os robôs a cabo permitem melhorias significativas no tamanho do espaço de trabalho, na carga máxima e nas capacidades dinâmicas devido à sua eficiência superior na transmissão de forças através de cabos quando comparados aos sistemas convencionais com elos rígidos [12].

Recentemente os projetos de robôs a cabos extrapolaram as paredes dos laboratórios e se encaixaram em diversas aplicações, como por exemplo a manipulação de materiais industriais em larga escala, logísticas internas e até algumas áreas da saúde como a fisioterapia [12]. Nas subsecções seguintes serão apresentados alguns grupos de estudos da área e as principais características dos robôs por eles desenvolvidos.

### 2.2.1 The NIST RoboCrane

Desenvolvido pelo Dr. James S. Albus, no Instituto Nacional de Padrões e Tecnologia dos EUA (NIST), o projeto teve início no começo dos anos 90. Albus deu início a uma nova geração de robôs a cabos, utilizando uma configuração básica do manipulador de elos paralelos da plataforma Stewart, os cabos desempenharam a função dos elos paralelos e os guinchos elétricos os atuadores, isto tornou possível o controlo fino dos seus seis graus de liberdade do sistema ( $x$ ,  $y$ ,  $z$ ,  $pitch$ ,  $roll$  e  $yaw$ ) [13].

Suas características baseiam-se em conectar a flange do robô a uma plataforma Stewart por meio de elos paralelos. Como pode ser visto na Figura 2.1, seus elos são cabos flexíveis não elásticos, conectados numa ponta a guinchos elétricos, e na outra ponta à flange do robô. Quando os seis cabos estão sob tensão, a plataforma é cinematicamente restringida, e pode-se aplicar uma relação matemática, sobre os comprimentos dos cabos, a qual dá o posicionamento e orientação da plataforma [14] [15].

O sistema utiliza um *joystick* para fazer o controlo do posicionamento da plataforma. Colocando os potenciômetros lineares do *joystick* na mesma posição que os cabos do robô, foi possível um sistema mestre-escravo. Neste modelo de controlo a velocidade de cada guincho (atuador) é dada pelo deslocamento do seu correspondente potenciômetro. Os potenciômetros do *joystick* geram tensões analógicas que passam por um conversor analógico/digital e são enviadas a um computador. Nos guinchos são colocados codificadores que geram sinais de onda quadrada que variam sua frequência conforme a velocidade dos motores. Sensores



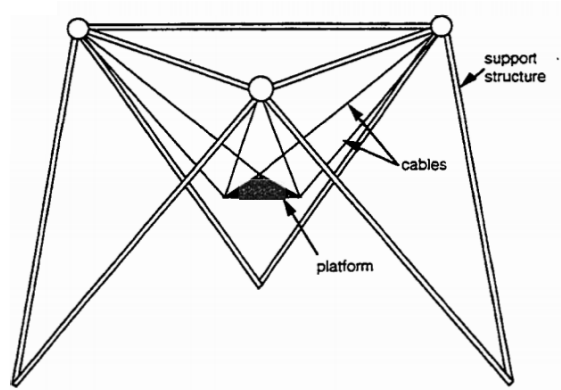


Figura 2.1: Estrutura básica do RoboCrane [15]

de força também foram instalados nos cabos para uma melhor realimentação em malha fechada do sistema. A Figura 2.2 representa o diagrama de controlo do sistema descrito acima [15].

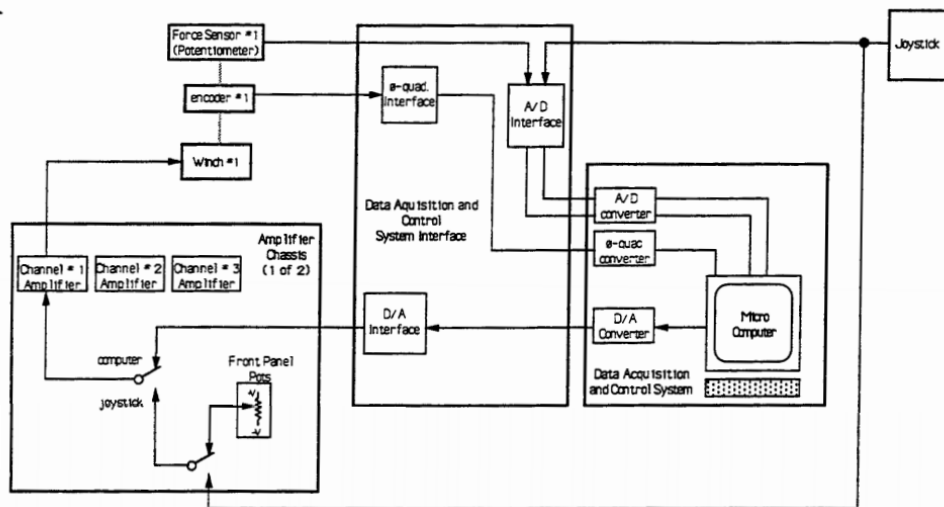


Figura 2.2: Sistema de controlo do The NIST RoboCrane [15]

O RoboCrane possui aplicações aéreas, aquáticas e espaciais. Possui versões com 2 m, com 6 m e outras foram criadas para aplicações específicas e analisado o seu desempenho para situações críticas.

Foram exploradas diferentes áreas para a aplicação, como pode ser visto na Figura 2.3. Duas aplicações distintas partindo do mesmo princípio, dentre elas estão: manutenção de aeronaves [16], grua expedicionária transportável aérea

[17], construção de ponte [18], expedição lunar [19], maquinação [20], manuseio de carga em navios [21], construção naval [22], configurações submarinas [23], combate a incêndios em crude em poços de petróleo [24].

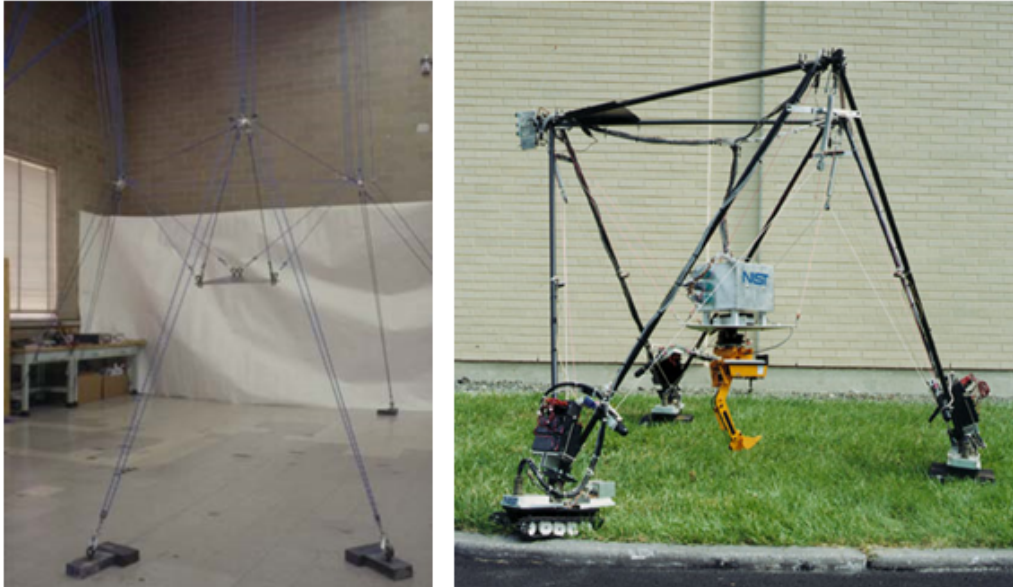


Figura 2.3: Configuração submarina (esquerda) [23] e expedição lunar (direita) [19]

### 2.2.2 Cable-Driven Parallel Robots IPAnema

A equipa Fraunhofer IPA começou os seus trabalhos em 2006 com o intuito de desenvolver uma família de robôs para inspeção, manuseio e montagens de média a grande escala. A família de robôs IPAnema é composta por três tipos de robôs a cabo. O primeiro é um sistema robótico modular com, até, oito cabos. O segundo é um robô planar de tamanho médio com quatro cabos e movimentos somente em duas dimensões. O terceiro é um sistema com seis cabos, com codificadores e sensores de comprimento do cabo que permitem medir a posição da plataforma no interior do volume de trabalho [11].

Os motores presentes nos guinchos são ligados aos controladores através de uma interface SERCOS ou EtherCAT, sendo os valores de *set-point* transferidos para o controlador com um tempo de ciclo de 2 ms. O sistema de barramento e o núcleo do controlador numérico podem alcançar uma frequência de *clock* de até 10 kHz, onde alguns dos algoritmos de controlo específicos para os sistemas robótico a cabos atualmente utilizados não são processados em tempos suficientemente rápidos para fornecer os dados em intervalos de tempo tão pequenos.

O sistema de controlo e os amplificadores de acionamento podem comandar a posição, velocidade e os valores de binário dos atuadores [11].

Duas gerações de robôs foram desenvolvidas pela equipa. A primeira geração, denominada de IPAnema 1, foi construída a partir de uma estrutura de alumínio, tinha capacidade de carga de até 3 kg, velocidade máxima de 10 m/s e aceleração máxima de 100 m/s<sup>2</sup>. O IPAnema 2 foi o responsável pela segunda geração, seu volume de trabalho e carga útil foram aumentadas por fatores de 10, chegou à dimensão de 8 m × 6 m × 5 m, e a carga útil passou a ser de 30 kg, porém a velocidade e a aceleração reduziram para 2,5 m/s e 25 m/s<sup>2</sup>, respetivamente. Os cabos, controladores e guinchos continuaram sendo os mesmos da primeira geração [11]. Na Figura 2.4 são apresentadas as duas gerações da família IPAnema de robôs a cabos.



Figura 2.4: IPAnema 1 (esquerda) e IPAnema 2 (direita) [11]

O núcleo do sistema de controlo (NC-kernel) é implementado em um computador com sistema operativo. Utilizando um computador industrial para fazer o controlo, é possível alcançar dois pontos muito importantes: a fiabilidade do sistema e a velocidade. Acoplado ao núcleo do sistema de controlo há o *software* de um controlador lógico programável (PLC). Para a geração da trajetória e para permitir supervisão e correção dos valores de posição do atuador, é necessário calcular a sua posição pelo comprimento dos cabos, resultando nas coordenadas cartesianas do flange do robô dadas em tempo real. As coordenadas cartesianas são calculadas através de um *software* de controlo numérico (NC) escrito em código G. Ao mesmo tempo são medidos os valores reais por meio de codificadores absolutos dentro dos guinchos, que são transformados na posição atual do atuador final. Para garantir o funcionamento correto do sistema robótico, a tensão nos cabos é monitorada, e isto pode ser feito de duas maneiras diferentes: a primeira delas é pela corrente do motor, a segunda e mais precisa, é a medição por meio de

sensores de força interligados nos pontos de ancoragem dos cabos [11]. A Figura 2.5 mostra a estrutura do sistema robótico apresentado acima.

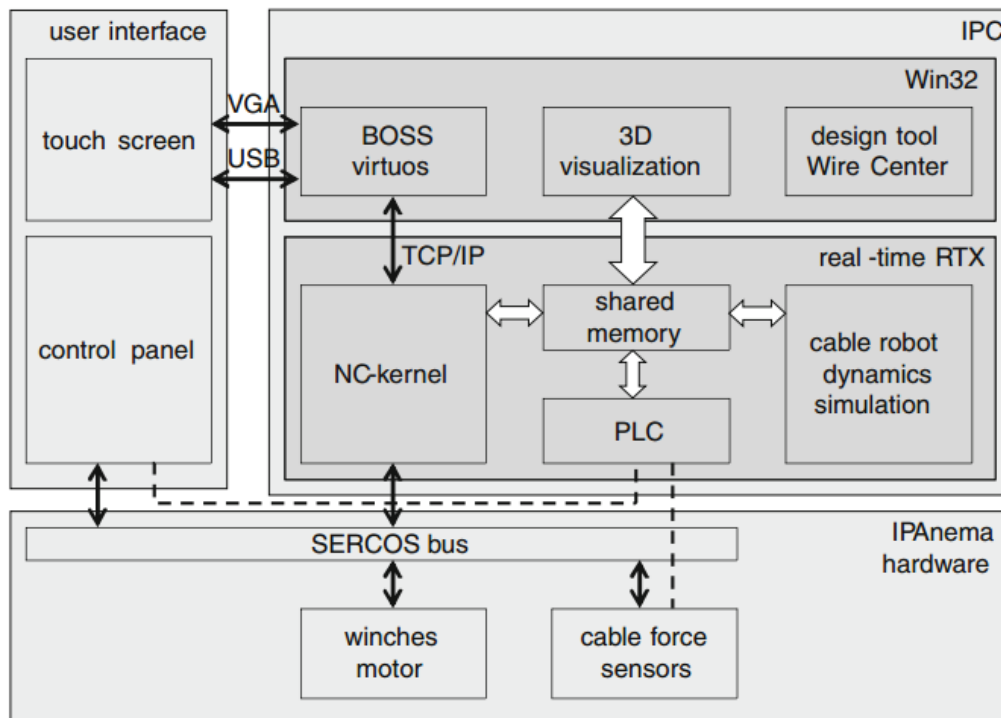


Figura 2.5: Estrutura do sistema de controle para o robô a cabos IPAnema [11]

### 2.2.3 REELAX: Configurable Robots of Small Size

Desenvolvido pela equipa liderada por Jean-Baptiste Izard, nas instalações da Tecnia France, este robô tem como principal característica a reconfiguração, sendo possível isto para a quantidade de cabos que sustentam a plataforma e para o tamanho do volume de trabalho [25] [26].

As especificações da geometria do robô baseiam-se no posicionamento dos pontos de saída dos cabos. As exigências da geometria do robô são básicas: ele deve ser reconfigurável, devem ser facilmente modificados os pontos de saída e os pontos de fixação, bem como o ponto de união dos cabos. Outro elemento que é levado em consideração é a facilidade em transportar os elementos do robô [26].

O número máximo de cabos para o sistema é oito, e o volume de trabalho é limitado ao comprimento total de cada cabo. Neste caso o comprimento é fixado para 6,6 m para poder atravessar a diagonal de um cubo de 4 m, que é suficiente para o máximo volume de trabalho do modelo mais pequeno de robô com dimensões 4 m × 3 m × 3 m [25] [26].

A carga máxima estipulada para este robô é de 25 kg, e os cabos devem ser capazes de suportar o peso total da plataforma com uma margem - para isto foi estipulado uma força de tração de 300 N. A velocidade máxima estimada é de 1,5 m/s [26]. A configuração do robô é baseada em quatro pilares de aço de 3 m de altura, sendo dois cabos presos a cada poste, como se pode ver na Figura 2.6.

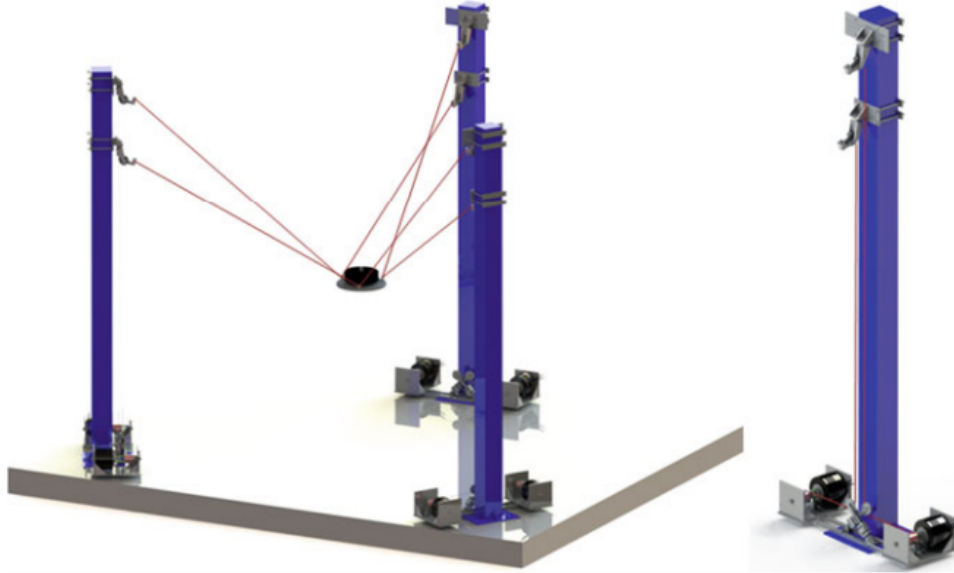


Figura 2.6: Vista lateral do ReelAx [26]

Com estes quatro postes a reconfiguração é feita tipicamente em três etapas: a primeira deve selecionar o número e a posição dos postes ao redor do volume de trabalho, o tamanho máximo do volume de trabalho e a quantidade de cabos. O segundo ponto a ser analisado é a posição do ponto de saída: este pode ser alocado em qualquer posição no poste de fixação, e o usuário é livre para escolher o espaçamento entre os pontos de saída, desde que estejam dentro de um intervalo pré-estabelecido. O terceiro parâmetro é a plataforma que será utilizada: ela é construída com o propósito de ter interfaces para a conexão com os cabos. Após serem analisados estes fatores pode-se configurar o sistema para a devida aplicação [26]. Na Figura 2.7 pode-se observar a configuração para quatro pilares, utilizando oito pontos de saída em alturas diferentes.

Os guinchos são colocados na base dos postes, como pode ser visto nas Figuras 2.6 e 2.7, e não podem ser movidos. Os cabos saem sempre na mesma posição dos atuadores e se dirigem a polias instaladas na parte superior do pilar, sendo estes os pontos de saída dos cabos. O robô é controlado usando um controlador MathWorks xPC Target. Suas leis de controle foram projetadas utilizando o MATLAB e o Simulink. O controlador do robô é conectado aos bornes utilizando

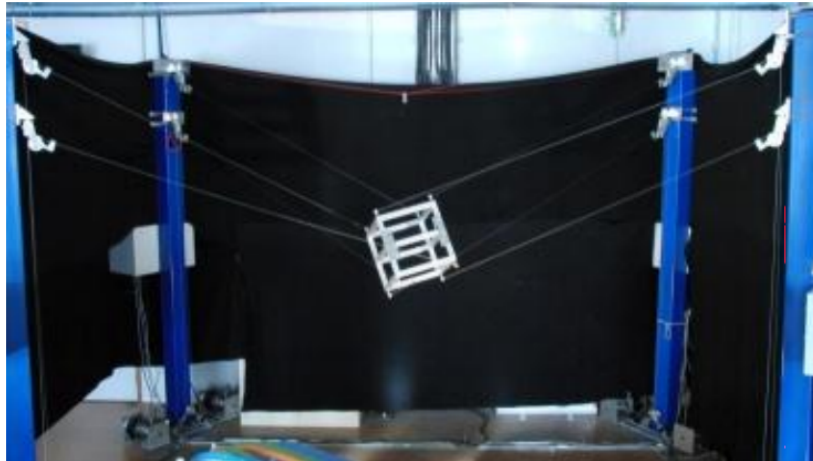


Figura 2.7: Vista lateral do ReelAx utilizando quatro pilares [25]

barramento Ethernet em tempo real, com base no protocolo de EtherCAT, ao qual as informações dos codificadores também são conectadas. Utilizando este protocolo as informações podem percorrer até 100 m. Cada poste é equipado com um acionamento AccurET da ETEL Motion Technology, que pode controlar os dois guinchos do pilar. Os atuadores são servomotores de acionamento direto sem escovas. Acoplado aos servomotores há um tambor onde é enrolado o cabo. Conforme pode ser visto na Figura 2.8, há um dispositivo mecânico para que o enrolamento do cabo seja de maneira correta e não haja erro na dinâmica da plataforma [26].

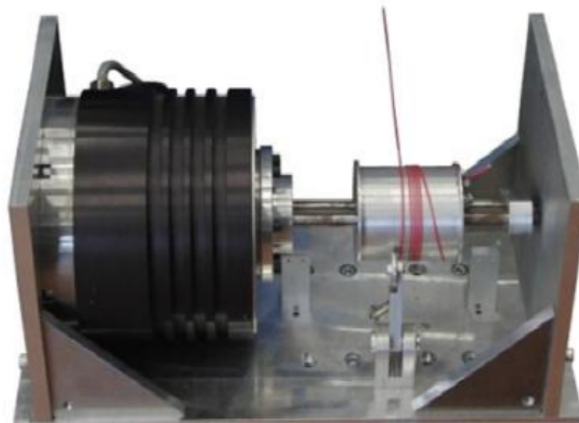


Figura 2.8: Servomotor com tambor para enrolamento acoplado [26]

Análogo ao ReelAx, a Tecnia desenvolveu alguns outros modelos de robôs com outras dimensões; dentre eles estão o HPRcable e o COGIRO. O HPRcable



foi projetado para pendurar um robô humanoide, como pode ser visto na Figura 2.9. Com um volume de trabalho médio com dimensões de  $11\text{ m} \times 4\text{ m} \times 3\text{ m}$  e oito cabos, sustenta até 60 kg [25].

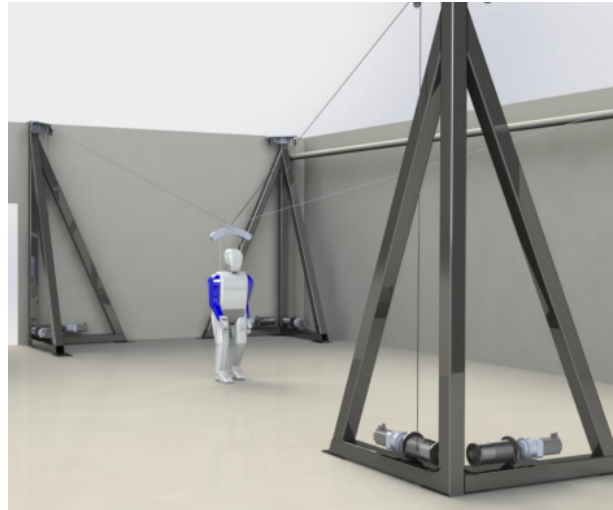


Figura 2.9: Vista frontal do HPRcable [25]

O COGIRO é um robô sem equivalente na Europa, com um volume de trabalho com dimensões de  $15\text{ m} \times 11\text{ m} \times 6\text{ m}$ , e uma capacidade de carga de 500 kg sendo o maior robô a cabos da Europa. Utiliza cabos de aço de 4 mm de diâmetro e um controlador desenvolvido pela B&R Automation. Possui uma precisão de posicionamento média de 50 mm e uma repetibilidade de posicionamento média de 3 mm. Este sistema pode ser visto na Figura 2.10 [25].



Figura 2.10: Vista frontal do sistema robótico COGIRO [27]

### 2.2.4 CABLEBOT – Maintance and Logistics of Large Scale Products

Com início no ano de 2011, o grupo de pesquisa denominado CableBOT, recebeu um financiamento de 3 milhões de euros da união europeia, os quais cobriram 66% do valor total do projeto (4,4 milhões de euros) [28].

O principal objetivo da equipa é desenvolver uma nova geração de dispositivos robóticos modulares e reconfiguráveis, capazes de realizar diferentes tarefas em diferentes etapas no ciclo de vida de um produto de grandes dimensões [28].

Três tecnologias básicas foram desenvolvidas para alcançar os objetivos esperados:

- Projeto do robô: ferramentas de *software* para projetar o *layout* dos robôs, ou seja, quais guinchos serão utilizados para as diferentes configurações do robô. Os grupos de guinchos escolhidos se comunicam com o circuito de controlo utilizando redes Wi-Fi não estruturadas (Ad-Hoc), a fim de obter uma maior flexibilidade e configurabilidade.
- Planejamento de processos industriais: simulação de robôs de cabo para verificar o funcionamento dos robôs em ambientes de larga escala.
- Algoritmo de controlo e sistemas: controlo distribuído e transformações cinemáticas para operar robôs de cabo modulares.

Duas principais aplicações foram apontadas pelo grupo de desenvolvimento, como pode ser visto na Figura 2.11. Uma delas é a manutenção e reparos em uma aeronave. Outra aplicação seria o manuseio de vigas utilizadas na construção civil [28].

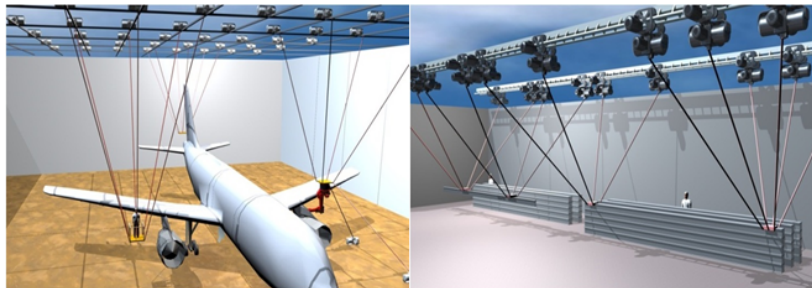


Figura 2.11: Reparos em aeronave (esquerda) e manuseio de vigas (direita) [28]

## 2.3 Conclusão do Capítulo

Conforme visto neste capítulo, desde a primeira aplicação na indústria os robôs só estiveram a ganhar mais espaço, chegando hoje em dia a atuar em áreas inimagináveis quando concebidos. Para poder alcançar tanto mercado os robôs



sofreram uma forte evolução, que vem desde sua estrutura mecânica, passando pelos sensores até os circuitos de controlo.

Tendo conhecimento da importância dos dispositivos para a evolução dos robôs, o capítulo seguinte é destinado aos atuadores e sensores para os robôs a cabos.

## Capítulo 3

---

# Atuadores e Sensores para Robôs a Cabos

---

*Na família de robôs a cabos os atuadores recebem o nome de guinchos, e são constituídos basicamente por dois tipos essenciais de motores. Já os sensores utilizados nos robôs a cabos não recebem uma denominação exclusiva pois são utilizados sensores de uso industrial comuns e estes atendem às necessidades dos sistemas robóticos. Abaixo será apresentada detalhadamente as características de cada um deles.*

### 3.1 Atuadores

Um atuador é um dispositivo capaz de transformar energia em movimento. Existem três principais tipos de atuadores: pneumáticos, elétricos e hidráulicos. Os robôs a cabos utilizam os atuadores elétricos, chamados de guinchos, por sua facilidade de controle e por não precisarem da infraestrutura que os hidráulicos e pneumáticos necessitam. Os guinchos elétricos podem ser movidos tipicamente por dois tipos principais de motores: os motores de passo e os servomotores com redutores e codificadores. Tipicamente os motores de passo possuem um binário mais baixo que os motores convencionais, por isso são utilizados para aplicações com carga mais reduzida.

#### 3.1.1 Guinchos

Recebem o nome de guinchos os atuadores dos robôs a cabos, e estes são compostos basicamente por três partes: o motor, o tambor e o codificador, podendo também ser acrescentado um sistema de guia para o cabo. Nos casos onde se utilizam motores de passo podem ser dispensados os codificadores nos guinchos,

porém uma boa prática é estar em algum ponto do sistema para que haja uma boa realimentação para o sistema de controlo. A Figura 3.1 representa o guincho utilizado nos robôs da família IPAnema, e nesta podem ser vistos estes componentes.

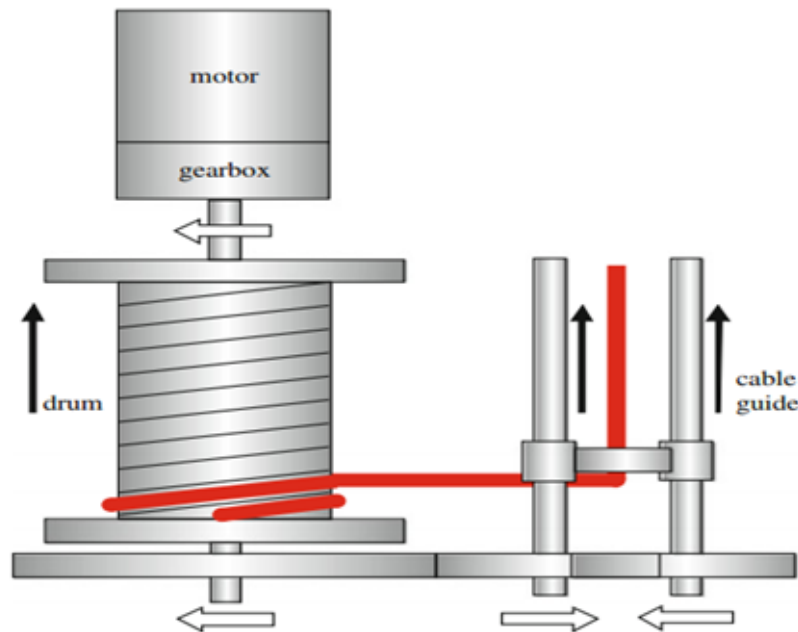


Figura 3.1: Esquema mecânico do guincho utilizado na família de robôs IPAnema [11]

O *design* do tambor tem grande importância em um guincho, pois as dimensões dele vão estipular a capacidade de cabos que o sistema comporta, e o quanto maior for o seu diâmetro maior será o comprimento do cabo dispensado ou recolhido em um pequeno deslocamento do eixo do motor. O sistema de guia do cabo é muito importante para o controlo de posição: o cabo sendo enrolado sempre na mesma posição e junto ao tambor, não há hipótese de o deslocamento do cabo ser maior que o esperado. Outro fator importante é a conservação do cabo: se o cabo se acomoda no tambor, o desgaste por atrito com outro por cima ou nas laterais é reduzido.

Conectado ao tambor há um motor, e este pode ser de dois tipos basicamente, os quais serão tratados nas subsecções seguintes.

### 3.1.2 Motores de Passo

Os motores de passo são motores de corrente contínua (CC) construídos tirando proveito da teoria das máquinas de relutância variável (MRV). A MRV tem um

funcionamento diferente dos motores convencionais, pois o deslocamento do seu eixo não é contínuo, e sim em etapas (passos). A MRV gira por um ângulo específico a cada passo, e o tamanho deste ângulo depende da quantidade de passos por revolução que possui o motor. Frequentemente os motores são projetados para possuir um grande número de passos por revolução, chegando até a 400 dentes no rotor com um espaçamento de  $0,9^\circ$  entre passos [29].

Um motor de passo é constituído basicamente de duas partes, o estator e o rotor. O estator é a parte fixa do motor, e é nela que estão os pares magnéticos formados pelos eletroímãs. O rotor é a parte móvel e fica no interior do estator. O rotor é construído com material magnético. Na Figura 3.2 pode-se observar estas duas partes do motor.

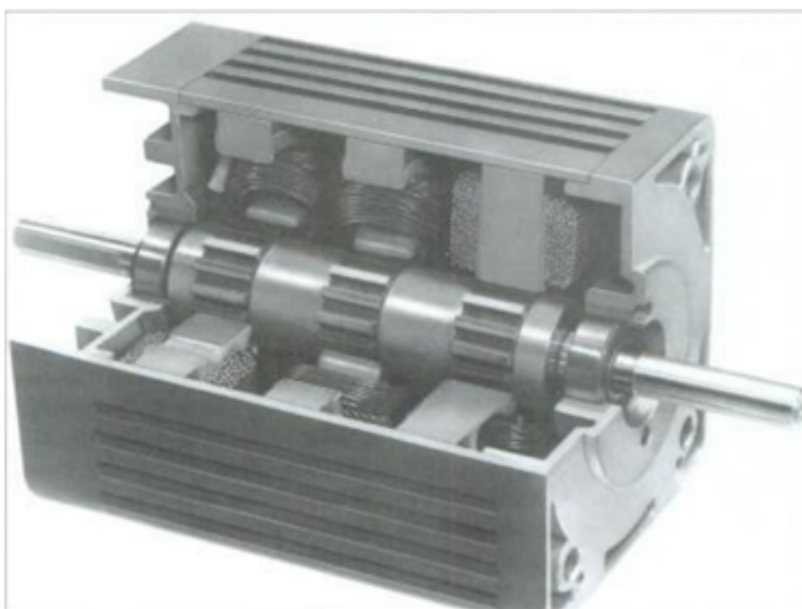


Figura 3.2: Vista em corte longitudinal de um motor de passo [29]

Quando o par magnético é energizado o rotor é atraído até ele, deslocando-se até o próximo passo. Estas características tornam o motor a passos compatível com os sistemas eletrônicos digitais, pois a cada impulso elétrico enviado o motor dá um passo, e assim é possível a sua utilização em aplicações com controlo de posição em sistemas de malha aberta. Em algumas aplicações a informação de posição pode ser obtida apenas com os registros do número de pulsos enviados ao motor: nestes casos o sistema de controlo é mais simples pois não necessita de sensores de realimentação [29].

### 3.1.3 Servomotor com Redutor e Codificador

Podem ser utilizados outros modelos de motores como atuadores dos robôs a cabos, porém estes devem sempre vir acompanhados de codificadores (o que torna mais complexo o circuito de controlo podendo vir a aumentar o erro do posicionamento do atuador final), e também de caixas redutoras de velocidade.

As caixas redutoras de velocidade, ou também conhecidas simplesmente como redutora, consistem em um conjunto de eixos com engrenagens cilíndricas de tamanhos diferentes acondicionadas dentro de uma carcaça lubrificada. Sua função é reduzir a velocidade de rotação do sistema à qual foi acoplado, e por consequência há um aumento significativo no binário transmitido. A parte fundamental em uma redutora são as engrenagens: o contacto entre os dentes das engrenagens de maior e menor diâmetro torna possível a redução da velocidade de rotação [30].

Acoplado ao redutor há um motor elétrico - que faz a conversão de energia elétrica em mecânica. Podem ser utilizados muitos tipos de motores para esta aplicação, porém o mais comum deles é o servomotor.

Os servomotores enquadram-se na classe de motores especiais, isto pelas suas características de controlo de velocidade e binário. Atualmente os servomotores são divididos em seis tipos: servomotor assíncrono de corrente alternada, servomotor síncrono de corrente alternada e servomotor de corrente contínua, servomotores de passo, servomotores de relutância chaveada e servomotores de indução[31].

Hoje em dia, com a evolução da eletrónica de potência, o servomotor mais utilizado é o servomotor síncrono de corrente alternada com ímanes no seu rotor, apresentando um menor volume para a mesma potência, com menor manutenção, pois em sua estrutura não apresenta coletor com escovas como nos servomotores de corrente contínua. Aliado a um bom sistema de controlo apresentam uma precisão ainda maior que a dos outros modelos [31].

As principais vantagens e desvantagens de cada tipo de motor para a aplicação tratada neste trabalho é apresentada na Tabela 3.1.

## 3.2 Sensores

Para este tipo de robô são utilizados dois tipos básicos de sensores: os codificadores, responsáveis por informar um posicionamento angular ou linear ao sistema de controlo, e os sensores de força, que realimentam o sistema com os valores das forças exercidas em cada cabo ou ponto de ancoragem do sistema.

Tabela 3.1: Motor de passo *vs* servomotor

Características	Motor de passo	Servomotor
Custo	Alto	Baixo
Binário	Baixo	Alto
Complexibilidade do circuito de controlo	Baixo	Alto
Necessidade de sensores de realimentação	Baixa	Alta
Necessidade de caixa de redução	Baixa	Alta
Tipo de tensão	CC	CA/CC

### 3.2.1 Codificador

Codificadores são os dispositivos capazes de converter o deslocamento angular ou linear, de um certo dispositivo mecânico, em um trem de pulsos facilmente interpretável. A partir desta informação é possível que o controlador do sistema possa calcular a velocidade e aceleração do dispositivo a ser monitorado.

Há dois tipos de codificadores, o absoluto e o incremental: a principal diferença entre os dois é o ponto que eles utilizam de referência para efetuar a medida. O codificador incremental dá o deslocamento somente sobre um ponto de referência inicial, já o absoluto mede o deslocamento em relação a um ponto de referência interno ao dispositivo [32].

O codificador incremental angular pode utilizar dois princípios diferentes em sua construção, a condução elétrica, ou transmissão de luz. Os codificadores por condução elétrica apresentam estrias condutoras depositadas sobre um disco isolante que é conectado ao dispositivo a ser monitorado. As estrias são eletricamente conectadas a um potencial elétrico no verso do disco e o contacto dos pinos nas estrias faz com que o circuito seja ligado e a saída fique em nível lógico baixo 0; quando o pino está entre estrias o seu potencial é alto, e a saída fica com nível lógico 1. Os codificadores baseados em transmissão de luz têm um funcionamento similar, porém, ao invés de estrias, há furos nos discos, e ao invés de pinos há emissores e recetores de luz, estando cada um de um lado do disco, alinhados com os furos; conforme há receção de luz, há mudança no sinal de saída [32].

O codificador absoluto tem um padrão convencional para os discos de codificação, consistindo este em uma série de trilhas com incrementos concêntricos, onde o número de ciclos por revolução dobra em cada trilha de raio crescente. Cada trilha tem seu próprio foto-detetor, e todos os detetores juntos formam um conjunto de bits dependendo da quantidade de pistas existentes, normalmente no código Gray [32].

### 3.2.2 Sensor de Força

Os sensores de força podem ser divididos em duas categorias: os quantitativos e os qualitativos. O sensor quantitativo realmente mede a força exercida naquele ponto, e representa seu valor em termos de sinais elétricos. Já os sensores qualitativos trabalham com um valor de referência e só emitem um sinal quando a força exercida no sensor excede esse valor, por exemplo [33].

Para a aplicação tratada neste trabalho podem ser utilizados ambos os tipos de sensores, porém o quantitativo oferece um melhor *feedback* para o sistema de controle.

A força pode ser medida de cinco maneiras diferentes. A primeira é equilibrar a força desconhecida contra a força gravitacional de uma massa padrão. A segunda é medir a aceleração de uma massa conhecida à qual a força é aplicada. A terceira equilibra a força com uma força desenvolvida electromagneticamente. A quarta converte a força em uma pressão de fluido e mede esta pressão. E, por fim, a quinta mede a deformação produzida em um elemento elástico pela força desconhecida [33].

Os sensores modernos utilizam geralmente a quinta maneira, e geralmente não convertem diretamente os valores das forças em sinais elétricos. Algumas etapas intermediárias são necessárias, e, por isto, os sensores de força são complexos. Por exemplo, como pode ser visto na Figura 3.3, um sensor pode ser construído a partir da combinação de dois elementos, sendo um de força ( $F$ ) *vs* deslocamento ( $x$ ), podendo ser uma mola helicoidal simples de constante elástica ( $k$ ) conhecida; e o outro um simples sensor de deslocamento ( $x$ ). A força ( $F$ ) é expressa a partir da Equação 3.1 [33].

$$F = k \times x \quad (3.1)$$

Neste caso particular, os sensores de força são colocados em série com os cabos ou pontos que desejam monitorar, e enviam sinais elétricos correspondentes aos valores mensurados para o sistema de controlo. Com o conhecimento destes valores há certeza que o sistema irá trabalhar sempre dentro da faixa de segurança, evitando uma possível sobrecarga nos motores, ou até o rompimento de um dos cabos.

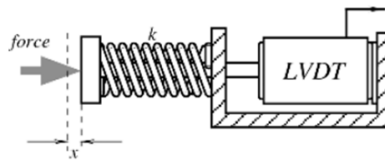


Figura 3.3: Sensor de mola com sensor de deslocamento [33]

### **3.3 Conclusão do Capítulo**

No decorrer deste capítulo foram apresentados os principais atuadores e sensores utilizados nos robôs atuados a cabos e quais as suas características. Tendo conhecimento de quais dispositivos podem ser utilizados, deve-se tomar conhecimento de como se dá a cinemática do robô, e este é o tema tratado no próximo capítulo.





## Capítulo 4

---

# Cinemática e Planeamento das Trajetórias dos Robôs Atuados por Cabos

---

*Os movimentos realizados pelos robôs são executados a partir do planeamento de uma trajetória. Para realizar o planeamento é necessário um cálculo geométrico utilizando ângulos e posições para estipular qual o caminho a ser realizado até ao destino. A cinemática é dividida em duas partes, a direta e a inversa, e cada uma delas visa obter resultados diferentes. Entende-se por cinemática direta quando se tem os ângulos e distâncias entre os elos do robô e, a partir disto, obtém-se a posição do atuador final. Este tipo de cálculo é muito importante para a simulação do sistema. Já a cinemática inversa é utilizada no sistema de controlo, e ela desempenha a função de informar, para os robôs paralelos atuados a cabos, o comprimento de cada cabo sabendo a posição do atuador final.*

### 4.1 Modelo no Plano $x - y$

A cinemática para robôs a cabos trabalha basicamente com três valores: os ângulos entre os cabos e os planos, o comprimento de cada cabo, e a posição espacial da flange do robô. Para facilitar o entendimento, nesta secção será apresentada somente a cinemática direta e inversa no plano  $x - y$  para um robô a quatro cabos.

Sua cinemática será resolvida com base na geometria analítica. A posição da flange do robô  $C = (x, y)$  e os pontos de fixação dos cabos, localizados em cada vértice  $(A_1, A_2, A_3, A_4)$ , são os parâmetros utilizados para encontrar os valores

de comprimento dos cabos ( $L_1, L_2, L_3, L_4$ ). A Figura 4.1 representa a estrutura simplificada no plano  $x - y$ .

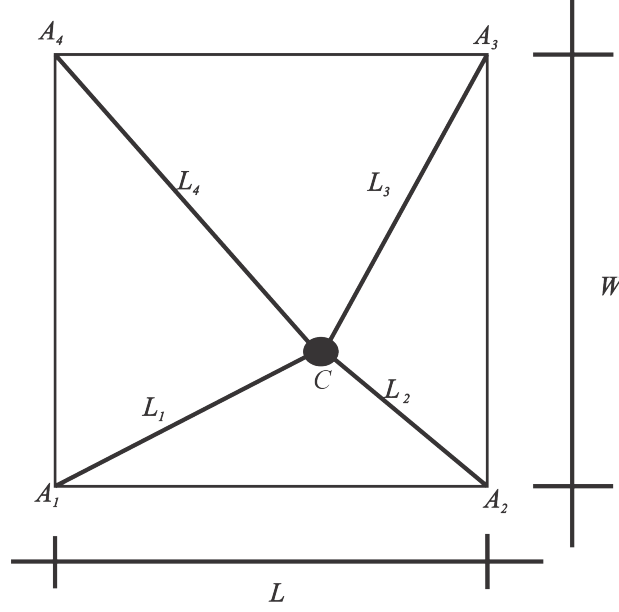


Figura 4.1: Modelo simplificado do sistema no plano  $x - y$

Para que o ponto  $C$  fique na posição desejada, os quatro cabos devem estar com os comprimentos iguais aos encontrados com a Equação 4.1.

$$L_i = \sqrt{(x - A_{i_x})^2 + (y - A_{i_y})^2}, \quad i = 1, \dots, 4 \quad (4.1)$$

Após obter o comprimento dos cabos é possível encontrar o valor dos ângulos entre o eixo positivo de  $x$  e o cabo. O valor correspondente pode ser expresso pela Equação 4.2.

$$\theta_i = \arctg_2 \frac{y - A_{i_y}}{x - A_{i_x}}, \quad i = 1, \dots, 4 \quad (4.2)$$

Quando se deseja encontrar a posição da flange do robô a partir do comprimento dos cabos é utilizada a cinemática direta. Para este sistema a cinemática direta é determinada a partir da intersecção de dois círculos, um centrado em  $A_1$  ( $0, 0$ ) e outro em  $A_2$  ( $L, 0$ ). O Sistema de Equações 4.3 representa a intersecção entre os dois círculos.

$$\begin{cases} (x - 0)^2 + (y - 0)^2 = L_1^2 \\ (x - L)^2 + (y - 0)^2 = L_2^2 \end{cases} \quad (4.3)$$

A partir da simplificação do Sistema de Equações 4.3 foram expressas as Equações 4.4 e 4.5, sendo estas as responsáveis por encontrar as coordenadas  $x$  e  $y$  do ponto  $C$ .

$$x = \frac{L^2 + L_1^2 - L_2^2}{2L} \quad (4.4)$$

$$y = \pm \sqrt{L_1^2 - x^2} \quad (4.5)$$

A Equação 4.5 expressa dois valores para a coordenada  $y$ , porém somente os valores positivos pertencem ao volume de trabalho do robô proposto neste trabalho, sendo considerado apenas o valor positivo.

Com o entendimento de como se dá a cinemática do movimento do robô, podem-se inserir as dimensões da plataforma nos cálculos e assim aproxima-se um pouco mais do modelo real. A Figura 4.2 expressa o sistema no plano  $x - y$  com a plataforma.

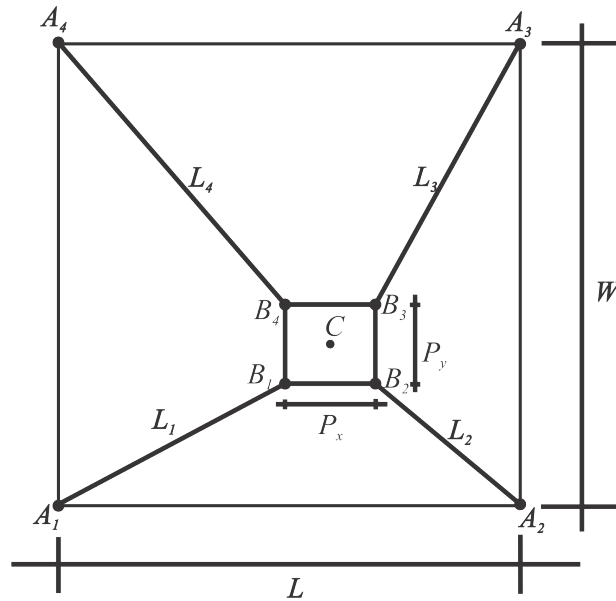


Figura 4.2: Modelo real do sistema no plano  $x - y$

Como pode ser visto na Figura 4.2, os cabos não estão mais ligados todos ao mesmo ponto, e com isto, as equações da cinemática inversa devem ser expressas levando em consideração estes fatores. As Equações 4.6 e 4.7 expressam as coordenadas  $x$  e  $y$  dos pontos de ancoragem dos cabos ( $B_i$ ) levando em consideração as dimensões da plataforma ( $P_x$  e  $P_y$ ) no plano  $x - y$ .

$$B_{i_x} = \begin{cases} x - \frac{P_x}{2}, & i = 1, 4 \\ x + \frac{P_x}{2}, & i = 2, 3 \end{cases} \quad (4.6)$$

$$B_{i_y} = \begin{cases} y - \frac{P_y}{2}, & i = 1, 2 \\ y + \frac{P_y}{2}, & i = 3, 4 \end{cases} \quad (4.7)$$

A partir dos valores obtidos com as Equações 4.6 e 4.7 pode-se utilizar a Equação 4.8 para encontrar o comprimento de cada cabo para o ponto  $C$ .

$$L_i = \sqrt{(B_{i_x} - A_{i_x})^2 + (B_{i_y} - A_{i_y})^2}, \quad i = 1, \dots, 4 \quad (4.8)$$

## 4.2 Modelo em 3 Dimensões (3D)

O próximo passo para descrever toda a cinemática do sistema é acrescentar o eixo  $z$  no modelo real no plano  $x - y$ . Com base na Figura 4.3 as posições dos pontos de saída de cada cabo são escritas em relação ao espaço 3D, sendo estes pontos dados pelas seguintes coordenadas:

$$\begin{aligned} A_1(x, y, z) &= (0, 0, H) \\ A_2(x, y, z) &= (L, 0, H) \\ A_3(x, y, z) &= (L, W, H) \\ A_4(x, y, z) &= (0, W, H) \end{aligned}$$

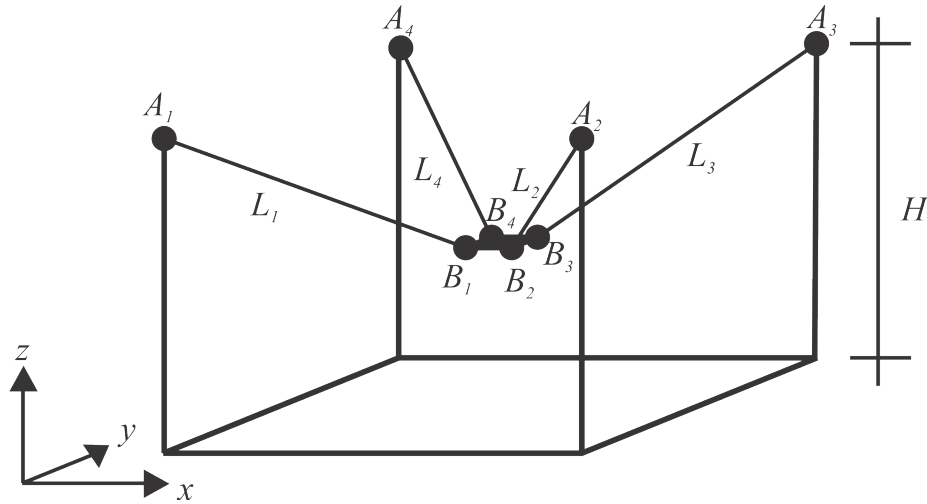


Figura 4.3: Modelo do sistema real em 3D com plataforma

### 4.2.1 Cinemática Inversa

Tendo conhecimento da estrutura do robô e das coordenadas dos pontos podem ser expressas as equações da cinemática inversa para o robô no espaço 3D. A Equação 4.9 expressa a cinemática inversa levando em consideração os pontos de ancoragem dos cabos na plataforma e os pontos de saída dos cabos na estrutura do robô.

$$L_i = \sqrt{(B_{i_x} - A_{i_x})^2 + (B_{i_y} - A_{i_y})^2 + (B_{i_z} - A_{i_z})^2}, \quad i = 1, \dots, 4 \quad , \quad (4.9)$$

onde  $B_{i_x}$ ,  $B_{i_y}$  e  $B_{i_z}$  são obtidos pelas Equações 4.6, 4.7 e 4.10, respectivamente.

$$B_{i_z} = \frac{P_z}{2}, \quad i = 1, \dots, 4 \quad , \quad (4.10)$$

sendo  $P_z$  o somatório das dimensões em  $z$  da garra mais a da plataforma.

### 4.2.2 Cinemática Direta

A cinemática direta é calculada para o espaço 3D por meio da intersecção de quatro esferas, onde os cabos são os raios de cada esfera, e os pontos  $A_1$ ,  $A_2$ ,  $A_3$  e  $A_4$  os respectivos centros. Com isto forma-se um sistema não-linear com quatro equações e três incógnitas. O Sistema de Equações 4.11 expressa uma aproximação para a cinemática direta de um robô paralelo atuado por quatro cabos.

$$\begin{cases} x^2 + y^2 + (z - H)^2 \approx (L_1 + M)^2 \\ (x - L)^2 + y^2 + (z - H)^2 \approx (L_2 + M)^2 \\ (x - L)^2 + (y - W)^2 + (z - H)^2 \approx (L_3 + M)^2 \\ x^2 + (y - W)^2 + (z - H)^2 \approx (L_4 + M)^2 \end{cases} \quad , \quad (4.11)$$

onde  $M$  é o segmento de reta entre o ponto de ancoragem dos cabos e o ponto  $C$ . A Equação 4.12 expressa como pode ser calculado o comprimento deste segmento.

$$M = \sqrt{\left(\frac{P_x}{2}\right)^2 + \left(\frac{P_y}{2}\right)^2 + \left(\frac{P_z}{2}\right)^2} \quad (4.12)$$

Para melhor entendimento, a Figura 4.4 representa graficamente os parâmetros presentes na Equação 4.12.

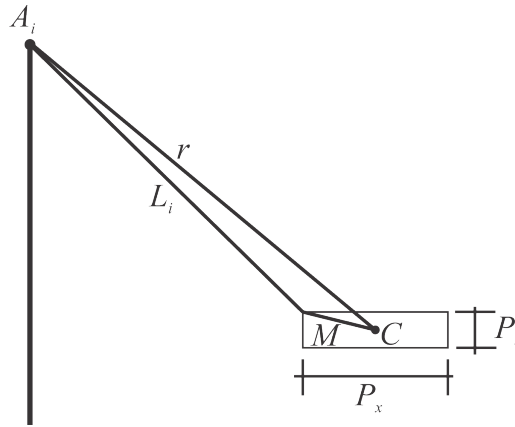


Figura 4.4: Representação gráfica do sistema utilizado para a aproximação da cinemática direta no plano  $x - y$

### 4.3 Orientação

Outros fatores ainda devem ser levados em consideração para descrever a cinemática da plataforma, estes fatores recebem o nome de orientação. Três parâmetros são observados na orientação de um sistema robótico, *Yaw*, *Pitch* e *Roll*, que em conjunto com as coordenadas  $x$ ,  $y$  e  $z$  completam os seis graus de liberdade do robô. A Figura 4.5 ilustra estes parâmetros.

A orientação deste sistema será tratada em duas partes, a orientação da plataforma e da garra. Nas subsecções seguintes serão tratados estes temas.

#### 4.3.1 Orientação da Plataforma

A orientação da plataforma tem uma grande importância para o sistema de controle, pois com orientações e comprimentos de cabo diferentes é possível obter o mesmo ponto para a ferramenta conectada a plataforma do robô, a Figura 4.6 ilustra esta situação.

Por se tratar de um robô atuado por quatro cabos, com a estrutura conforme a apresentada na Figura 4.3, não é possível que o *Roll* exerça influência significativa na posição da plataforma, portanto é desprezado dos cálculos.

Com isto o sistema passa a ter cinco graus de liberdade ( $x$ ,  $y$ ,  $z$ , *Yaw* e *Pitch*), se for levada em consideração a plataforma sem o atuador final. Considerando estas duas orientações, as Equações 4.6, 4.7 e 4.10 utilizadas para o cálculo das coordenadas dos pontos  $B_1$ ,  $B_2$ ,  $B_3$  e  $B_4$  não podem mais ser utilizadas para solucionar o problema, pois não apresentam as componentes trigonométricas que agora serão necessárias devido aos ângulos que surgem.

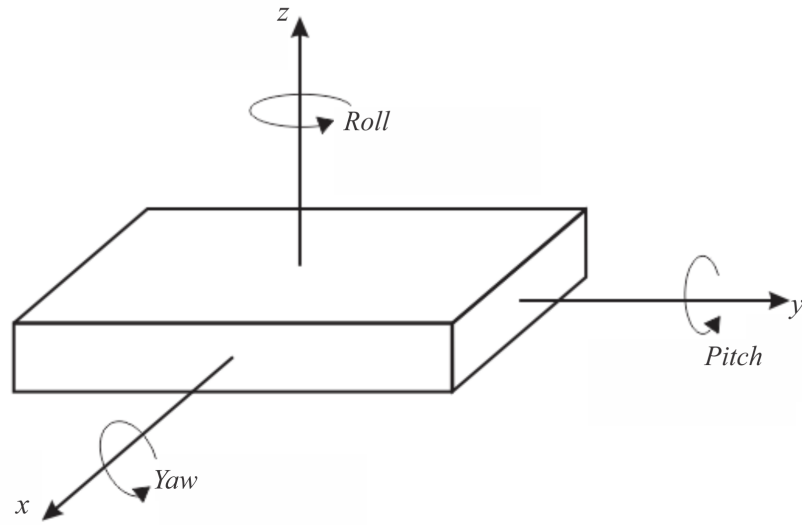


Figura 4.5: *Yaw*, *Pitch* e *Roll* na plataforma com referencial cartesiano

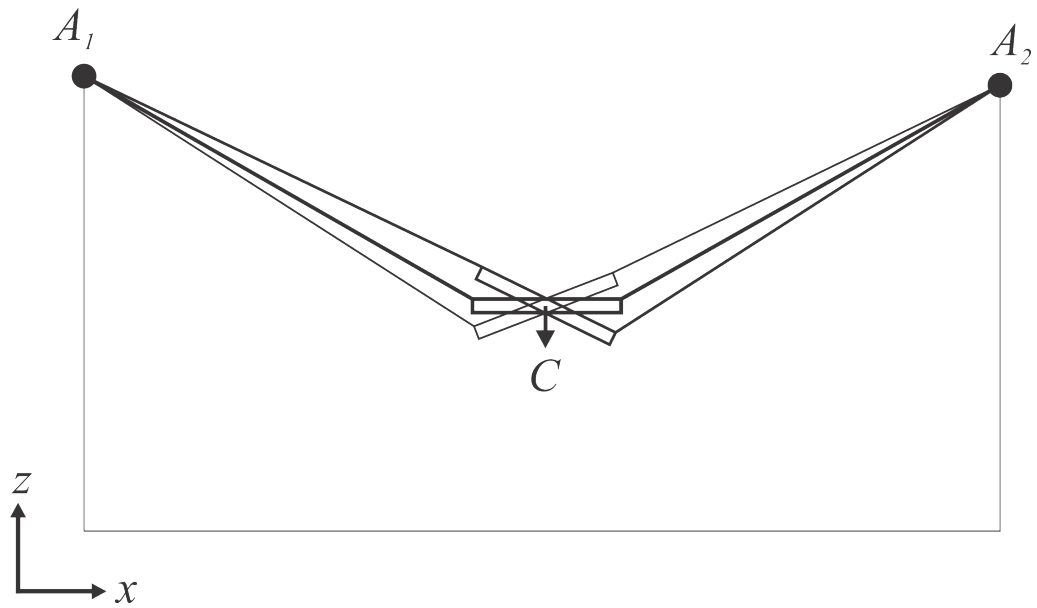


Figura 4.6: Sistema com diferentes orientações da plataforma e comprimentos de cabo para o mesmo ponto  $C$

Considerando uma rotação  $\alpha$  em *Pitch*, com o eixo de rotação no centro da plataforma, as coordenadas do ponto  $C$  não serão alteradas, porém os pontos  $B_1$ ,



$B_2$ ,  $B_3$  e  $B_4$  sofrerão alterações em suas coordenadas em relação a  $x$  e  $z$ . A Figura 4.7 representa esta variação de posição dos pontos de ancoragem do sistema.

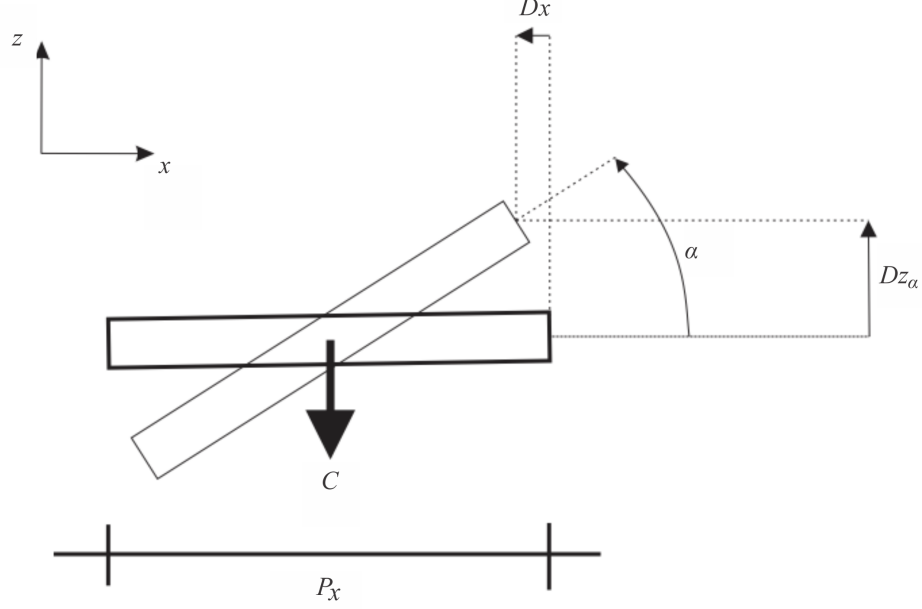


Figura 4.7: Rotação *Pitch* no sistema

Os valores de  $Dx$  e  $Dz_\alpha$  podem ser expressos segundo as Equações 4.13 e 4.14.

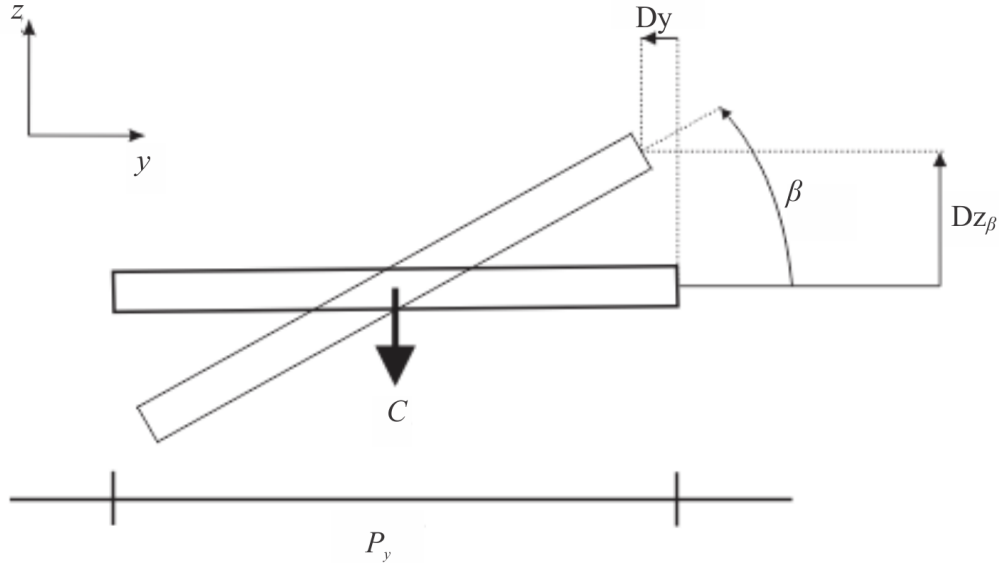
$$Dx = \frac{P_x}{2} \times (1 - \cos(\alpha)) \quad (4.13)$$

$$Dz_\alpha = \frac{P_x}{2} \times \sin(\alpha) \quad (4.14)$$

Com variações nas coordenadas em  $x$  e  $z$  a Equação 4.9 não pode mais ser utilizada para encontrar o comprimento dos cabos, devendo-se utilizar a Equação 4.15 para este cálculo.

$$L_i = \sqrt{[(B_{i_x} + Dx) - A_{i_x}]^2 + [B_{i_y} - A_{i_y}]^2 + [(B_{i_z} + Dz_\alpha) - A_{i_z}]^2}, \quad i = 1, \dots, 4 \quad (4.15)$$

Similar à rotação em *Pitch*, a rotação em *Yaw* também causa efeito nas coordenadas dos pontos  $B_1$ ,  $B_2$ ,  $B_3$  e  $B_4$ , porém desta vez em relação a  $y$  e  $z$ . A Figura 4.8 representa o movimento da plataforma considerando um ângulo  $\beta$  em *Yaw* à plataforma com o eixo de rotação no centro da plataforma.

Figura 4.8: Rotação *Yaw* no sistema

Os valores de  $Dy$  e  $Dz_\beta$  podem ser expressos segundo as Equações 4.16 e 4.17.

$$Dy = \frac{P_y}{2} \times (1 - \cos(\beta)) \quad (4.16)$$

$$Dz_\beta = \frac{P_y}{2} \times \sin(\beta) \quad (4.17)$$

Assim como a rotação em *Pitch*, para a rotação em *Yaw* a Equação 4.9 não pode ser utilizada para calcular o comprimento dos cabos. A Equação 4.18 deve ser utilizada para solucionar este problema.

$$L_i = \sqrt{[B_{i_x} - A_{i_x}]^2 + [(B_{i_y} + Dy) - A_{i_y}]^2 + [(B_{i_z} + Dz_\beta) - A_{i_z}]^2}, i = 1, \dots, 4 \quad (4.18)$$

Em uma situação real, no espaço 3D, pode-se obter uma rotação em torno de  $x$  e  $y$  simultaneamente, esta situação exerce influência nas coordenadas dos pontos  $B_1$ ,  $B_2$ ,  $B_3$  e  $B_4$ . Por questões temporais não será trabalhada a matemática para este caso nesta obra.

#### 4.3.2 Orientação da Garra

A orientação da garra é feita de forma diferente da plataforma, pois só são trabalhados quatro graus de liberdade ( $x$ ,  $y$ ,  $z$  e *Roll*). Para que não haja *Pitch*

e *Yaw* na garra é utilizado um dispositivo mecânico (cardã) que deixa a garra sempre orientada no sentido negativo do eixo  $z$ . O *Roll* é controlado por meio de um servo motor colocado na plataforma, com ele é possível que a garra assuma qualquer ângulo no plano  $x - y$ .

## 4.4 Planeamento das Trajetórias

Para que o robô realize um movimento, saia de um ponto  $P0$  e chegue a um ponto  $P1$ , por exemplo, a flange do robô passa por vários pontos intermediários, com uma velocidade e aceleração - a história temporal de cada um destes parâmetros recebe o nome de trajetória.

O planeamento das trajetórias é feito por meio de uma função matemática que descreve o comportamento do robô. Para que o deslocamento do robô seja suave, ou seja, em rampa e não degraus, é necessário que a função seja suave. Uma função suave é contínua, e tem a sua primeira derivada contínua; por vezes é necessário que a segunda derivada também seja contínua.

Sabendo que a equação descreve a posição do robô, partindo da física clássica, a primeira derivada descreve a velocidade e a segunda derivada a aceleração.

A trajetória deste robô é descrita por um polinómio de quinto grau, pois pretende-se especificar posição, velocidade e aceleração em todos os instantes de tempo de um segmento de um trajeto. Isto pode ser feito de duas maneiras: linear no espaço das juntas ou linear no espaço operacional [34]. Para este robô foi feito o estudo para ambos os casos e este estudo é apresentado nas subsecções a seguir.

### 4.4.1 Planeamento da Trajetória Linear no Espaço das Juntas

O planeamento das trajetórias é feito no espaço das juntas quando deseja-se que os motores efetuem seu movimento na mesma direção durante todo o movimento, porém quando optado por este tipo de trajetória não é garantido que o movimento aconteça em linha reta no espaço operacional. A Equação 4.19 representa o comprimento do cabo a cada instante de tempo durante a trajetória.

$$L_i(t) = a_{0_i} + a_{1_i} \times t + a_{2_i} \times t^2 + a_{3_i} \times t^3 + a_{4_i} \times t^4 + a_{5_i} \times t^5, \quad i = 1, \dots, 4 \quad (4.19)$$

Derivando a Equação 4.19 a fim de obter as velocidades e acelerações iniciais e finais obtém-se o Sistema 4.20 com seis equações e seis incógnitas.

$$\begin{cases} L_{ii} = a_{0_i} \\ L_{if} = a_{0_i} + a_{1_i} \times tf + a_{2_i} \times tf^2 + a_{3_i} \times tf^3 + a_{4_i} \times tf^4 + a_{5_i} \times tf^5 \\ \dot{L}_{ii} = a_{1_i} \\ \dot{L}_{if} = a_{1_i} + 2a_{2_i} \times tf + 3a_{3_i} \times tf^2 + 4a_{4_i} \times tf^3 + 5a_{5_i} \times tf^4 \\ \ddot{L}_{ii} = 2a_{2_i} \\ \ddot{L}_{if} = 2a_{2_i} + 6a_{3_i} \times tf + 12a_{4_i} \times tf^2 + 20a_{5_i} \times tf^3 \end{cases} \quad (4.20)$$

onde  $L_{ii}$  é o comprimento do cabo para a posição inicial e  $L_{if}$  é o comprimento do cabo para a posição final para  $i = 1, \dots, 4$  e  $tf$  é o tempo de duração do movimento.

Aplicando as restrições de velocidade e aceleração ao Sistema 4.20 chega-se ao Sistema 4.21 que expressa os valores dos coeficientes do polinómio apresentado na Equação 4.19.

$$\begin{cases} a_{0_i} = L_{ii} \\ a_{1_i} = \dot{L}_{ii} = 0 \\ a_{2_i} = \frac{\ddot{L}_{ii}}{2} = 0 \\ a_{3_i} = \frac{20 \times (L_{if} - L_{ii})}{2 \times tf^3} \\ a_{4_i} = \frac{30 \times (L_{ii} - L_{if})}{2 \times tf^4} \\ a_{5_i} = \frac{12 \times (L_{if} - L_{ii})}{2 \times tf^5} \end{cases} \quad (4.21)$$

A Figura 4.9 apresenta os gráficos de deslocamento dos cabos  $L_1$ ,  $L_2$ ,  $L_3$  e  $L_4$ , resultando em um movimento no atuador final da posição inicial com coordenadas (1; 2; 1) para a posição final com coordenadas (1,25; 2; 1) em um tempo de 1 segundo.

Pelo fato do movimento no atuador final estar acontecendo no sentido positivo do eixo  $x$  os cabos  $L_2$  e  $L_3$  estão se deslocando no sentido oposto dos cabos  $L_1$  e  $L_4$ , por isto as velocidades e acelerações dos cabos  $L_2$  e  $L_3$  estão com os gráficos espelhados aos dos cabos  $L_1$  e  $L_4$ .

#### 4.4.2 Planeamento da Trajetória Linear no Espaço Operacional

O planeamento das trajetórias é feito no espaço operacional quando se deseja que o movimento seja realizado em linha reta, porém para garantir que isto ocorra os motores podem realizar diversas inversões de sentido durante o movimento. As Equações 4.22, 4.23 e 4.24 apresentam as coordenadas  $x$ ,  $y$  e  $z$ , respetivamente, em cada instante de tempo.

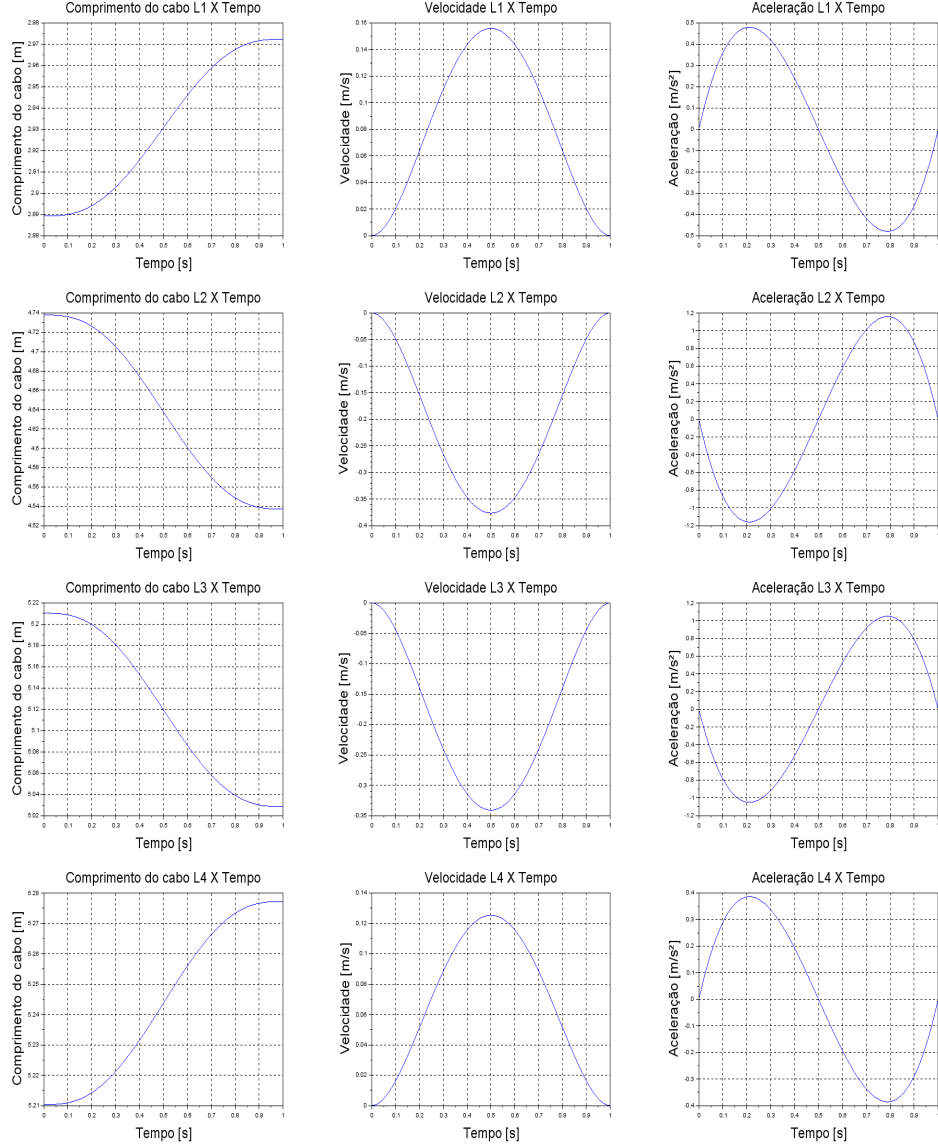


Figura 4.9: Evolução do comprimento (esquerda), velocidade (centro) e aceleração (direita) ao longo do tempo para os cabos  $L_1$  (primeira fileira),  $L_2$  (segunda fileira),  $L_3$  (terceira fileira) e  $L_4$  (última fileira)

$$x(t) = a_{0_x} + a_{1_x} \times t + a_{2_x} \times t^2 + a_{3_x} \times t^3 + a_{4_x} \times t^4 + a_{5_x} \times t^5 \quad (4.22)$$

$$y(t) = a_{0_y} + a_{1_y} \times t + a_{2_y} \times t^2 + a_{3_y} \times t^3 + a_{4_y} \times t^4 + a_{5_y} \times t^5 \quad (4.23)$$

$$z(t) = a_{0_z} + a_{1_z} \times t + a_{2_z} \times t^2 + a_{3_z} \times t^3 + a_{4_z} \times t^4 + a_{5_z} \times t^5 \quad (4.24)$$

De forma análoga ao espaço das juntas, os polinómios característicos do movimento foram derivados, obtendo as expressões de velocidade e aceleração; juntas, estas equações formam um sistema ao qual foram aplicadas as restrições de velocidade e aceleração chegando aos sistemas de equações 4.25, 4.26 e 4.27 que expressam os valores dos coeficientes dos polinómios apresentados nas Equações 4.22, 4.23 e 4.24, respetivamente.

$$\begin{cases} a_{0_x} = x_i \\ a_{1_x} = \dot{x}_i = 0 \\ a_{2_x} = \frac{\ddot{x}_i}{2} = 0 \\ a_{3_x} = \frac{20 \times (x_f - x_i)}{2 \times t_f^3} \\ a_{4_x} = \frac{30 \times (x_i - x_f)}{2 \times t_f^4} \\ a_{5_x} = \frac{12 \times (x_f - x_i)}{2 \times t_f^5} \end{cases} \quad (4.25)$$

onde  $x_i$  é a coordenada em  $x$  inicial e  $x_f$  a coordenada em  $x$  final.

$$\begin{cases} a_{0_y} = y_i \\ a_{1_y} = \dot{y}_i = 0 \\ a_{2_y} = \frac{\ddot{y}_i}{2} = 0 \\ a_{3_y} = \frac{20 \times (y_f - y_i)}{2 \times t_f^3} \\ a_{4_y} = \frac{30 \times (y_i - y_f)}{2 \times t_f^4} \\ a_{5_y} = \frac{12 \times (y_f - y_i)}{2 \times t_f^5} \end{cases} \quad (4.26)$$

onde  $y_i$  é a coordenada em  $y$  inicial e  $y_f$  a coordenada em  $y$  final.

$$\begin{cases} a_{0_z} = z_i \\ a_{1_z} = \dot{z}_i = 0 \\ a_{2_z} = \frac{\ddot{z}_i}{2} = 0 \\ a_{3_z} = \frac{20 \times (z_f - z_i)}{2 \times t_f^3} \\ a_{4_z} = \frac{30 \times (z_i - z_f)}{2 \times t_f^4} \\ a_{5_z} = \frac{12 \times (z_f - z_i)}{2 \times t_f^5} \end{cases} \quad (4.27)$$

onde  $z_i$  é a coordenada em  $z$  inicial e  $z_f$  a coordenada em  $z$  final.

A Figura 4.10 representa os gráficos de deslocamento do atuador final partindo da posição com coordenadas (1; 2; 1) e chegando à posição com coordenadas (1,25; 1,75; 2) referente aos eixos  $x$ ,  $y$  e  $z$ .

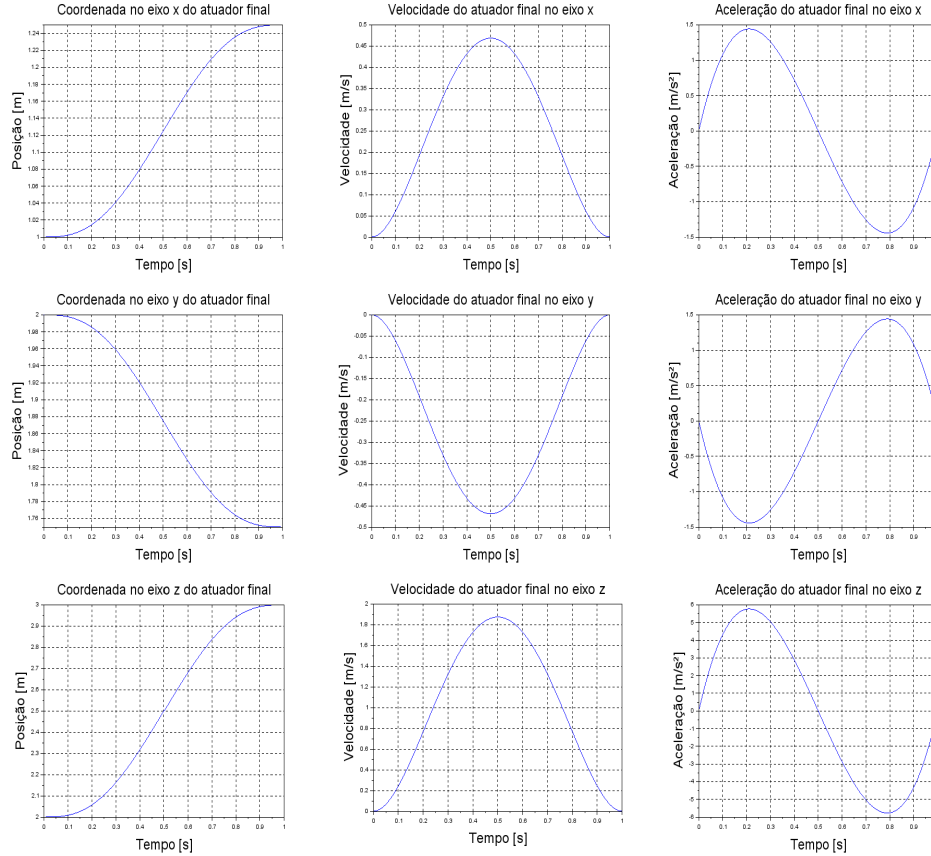


Figura 4.10: Evolução da posição (esquerda), velocidade (centro) e aceleração (direita) do atuador final em  $x$  (primeira fileira),  $y$  (segunda fileira) e  $z$  (última fileira) ao longo do tempo

## 4.5 Conclusão do Capítulo

Neste capítulo foi apresentado o conceito da cinemática direta e inversa para um robô atuado por quatro cabos e suas expressões características; foi apresentado também o conceito de orientação relativa, sendo esta dividida em duas partes para o robô desenvolvido neste trabalho: a orientação da plataforma e da garra. Foi apresentado o planeamento da trajetória, bem como as equações, e os gráficos gerados pela simulação do movimento.

Com o conhecimento de como se dá a cinemática do robô e quais os dispositivos devem ser utilizados, o próximo tema a ser trabalhado é a arquitetura do

sistema, e o capítulo seguinte contempla este tema.





## Capítulo 5

---

# Arquitetura do Sistema

---

*Os robôs podem ser de duas tipologias: série e paralelo. Dentre os paralelos podem haver diferentes arquiteturas sendo atuados de diferentes maneiras. O robô paralelo proposto neste trabalho apresenta uma arquitetura com quatro cabos e guinchos. Este capítulo apresenta os detalhes da arquitetura.*

### 5.1 Requisitos do Sistema Robótico

O sistema robótico que é proposto neste trabalho foi projetado para atender a alguns requisitos básicos para a aplicação desejada. Sendo esta a construção civil, uma das principais necessidades do sistema é a carga útil. Para este projeto foi estipulado uma carga útil máxima de 5 kg. Outro fator de igual importância é o volume de trabalho e, neste projeto, este parâmetro não é fixo, portanto alcançar-se-á o espaço de trabalho máximo com as dimensões de 5 m  $\times$  5 m  $\times$  3 m.

As dimensões do robô são dadas pela altura das torres (eixo  $z$ ) e pelo espaçamento entre elas (plano  $x - y$ ). Por se tratar de um sistema com as dimensões no plano  $x - y$  não fixas, o espaçamento entre as torres não é fixo, porém é limitado ao valor máximo de 5 m.

As torres devem estar todas no mesmo nível para que o sistema possa responder conforme ao esperado. Já a orientação relativa entre as torres não precisa ser controlada, desde que as roldanas na parte superior das torres consigam fazer a varredura dos 90° entre as torres vizinhas. É suposto que as torres estejam na base de um quadrilátero com ângulos internos de 90° com as roldanas voltadas para o interior do volume de trabalho.

A velocidade média estipulada para o atuador final deste robô é de 0,25 m/s.

## 5.2 Arquitetura do Sistema

Durante o estudo inicial, foram apontadas várias funcionalidades que suprissem os requisitos do sistema, algumas encontradas em robôs similares citados acima, outras que adequem da melhor forma o robô para as necessidades da sua aplicação específica, levando em consideração a hostilidade do ambiente onde estará inserido.

### 5.2.1 Estrutura

A principal aplicação deste robô é a construção civil, e para isto precisa de uma estrutura robusta, seja para suportar as cargas necessárias neste ambiente, bem como os possíveis choques com outras ferramentas do ambiente.

Sua sustentação é feita por meio de quatro pilares estaiados no solo, como pode ser visto na Figura 5.1, encontrando-se na base de cada um deles uma caixa que portará o guincho, e seus sensores. No interior de cada caixa há um motor, e ligado ao eixo deste, há um tambor para o cabo de aço, e este é responsável para que o cabo seja enrolado e desenrolado de forma correta para que não haja nenhum dano ao cabo, e também para que o erro na localização do atuador final seja o menor possível.

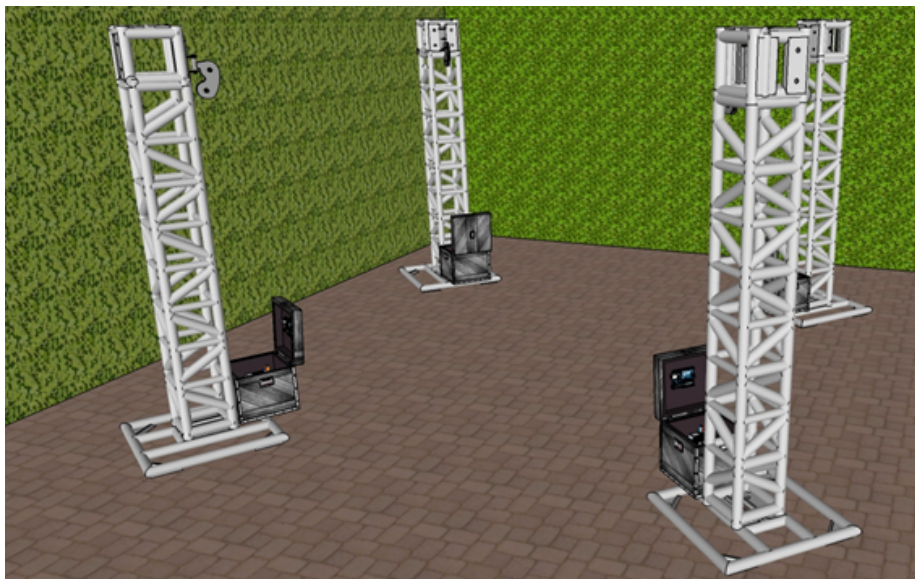


Figura 5.1: Modelo 3D da estrutura mecânica

Em cada caixa há somente uma entrada para a alimentação elétrica, e uma saída, que será o cabo conectado ao atuador final. No topo dos pilares há uma roldana fixada a um mancal para que ela possa se posicionar da melhor forma

para que a agressão ao cabo seja a menor possível. Entre a saída da caixa e a roldana há um cano de proteção, por dentro do qual passa o cabo, evitando que haja acidentes que rompam o cabo, ou até que possam machucar o utilizador.

Os pilares não terão uma distância pré-definida uns dos outros, pois acredita-se que nem todas as aplicações necessitam das mesmas condições, e quão mais dinâmico se torna o sistema, mais versátil ele será.

A ferramenta instalada no atuador final é uma garra, e como se trata de materiais com uma massa considerável, a garra terá um sensor fim de curso em seu interior, e somente quando ele for atuado a garra para de fechar. Como pode ser visto na Figura 5.2, no lado que opõe o sensor há uma borracha macia, a qual se adapta ao material a ser carregado e aumenta o atrito entre a peça e a garra. A orientação desejada na garra no plano  $x - y$  é feita por um servomotor instalado na plataforma. A conexão entre a garra e a plataforma é feita por meio de um dispositivo mecânico (cardã), como pode ser visto na Figura 5.2, que permite que a garra esteja sempre orientada na direção da gravidade, com isso não serão controlados todos os graus de liberdade.

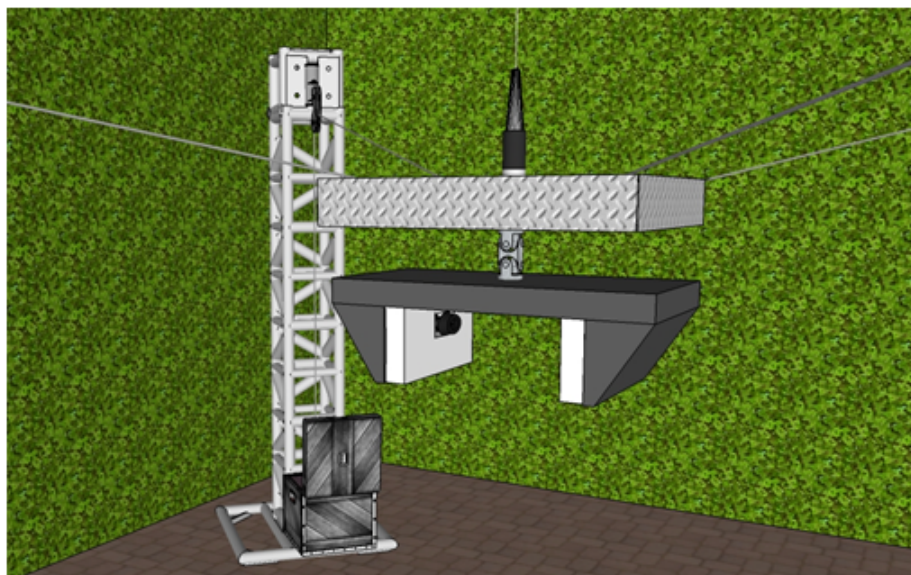


Figura 5.2: Atuador final conectado ao sistema

Nos pilares há um local para que sejam conectados os cabos de aço que estarão ligados às estacas fincadas no solo, é isso que permite que o atuador final manuseie as peças no interior do volume de trabalho e os pilares se tornem estáticos, e também, desta forma eles podem facilmente ser mudados de local.

### 5.2.2 Controlo

O núcleo do sistema de controlo (*kernel*) do dispositivo é feito em um computador portátil com sistema operacional, nele há um *software* com interface para o utilizador, onde há campos para o utilizador introduzir qual a distância entre os pilares de sustentação, e qual o próximo ponto do atuador, bem como a orientação em graus da garra no plano  $x - y$  e sua situação (aberta ou fechada).

O controlo individual de cada motor é feito por uma placa microcontrolada, equipada com periféricos capazes de comunicar via rádio (Wi-Fi) com o computador portátil (*kernel*). A placa também deverá ser capaz de interpretar os sinais recebidos pelos sensores e enviar para o *kernel* do sistema.

O computador envia pacotes de dados com as informações de controlo para as placas que estão na base dos pilares junto aos motores e sensores, e as placas são quem controla o quanto o motor deve girar, com base nas informações recebidas do *kernel*. Para que estes valores sejam exatos, há um sensor na saída da caixa que mede o comprimento do cabo que foi dispensado ou recolhido a partir do comando dado ao motor.

Para fazer o controlo da garra há uma placa de controlo no atuador final, e esta, a partir dos dados recebidos do *kernel*, envia o comando aos atuadores da garra. A Figura 5.3 ilustra as interações do sistema de controlo do robô.

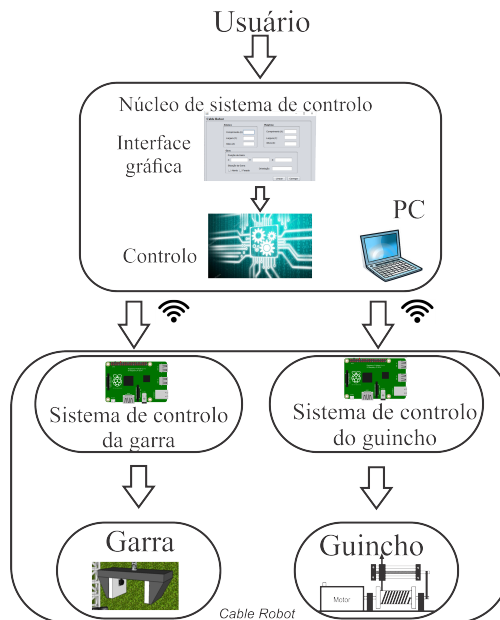


Figura 5.3: Representação das interações do sistema

### 5.2.3 Comunicação

A comunicação entre o computador portátil e as placas de controlo é feita utilizando o protocolo IEEE 802.11, para isto as placas de controle utilizam periféricos Wi-Fi. A rede utilizada para fazer a comunicação é criada a partir de um router. Cada periférico tem um endereço *Media Access Control* (MAC) e *Internet Protocol* (IP) estático, não precisando assim que o utilizador esteja a procurar os endereços dos dispositivos todas as vezes que o for utilizar.

Utilizando uma comunicação via rádio a versatilidade do dispositivo fica ainda maior, podendo ser facilmente transportado ou alterado o seu espaço de trabalho, pois como pode ser visto na Figura 5.4, não há o cuidado necessário com os cabos de controlo, sendo necessário apenas a alteração dos valores referentes às dimensões da estrutura do robô nos campos destinados a isto presentes no *software* de controlo.

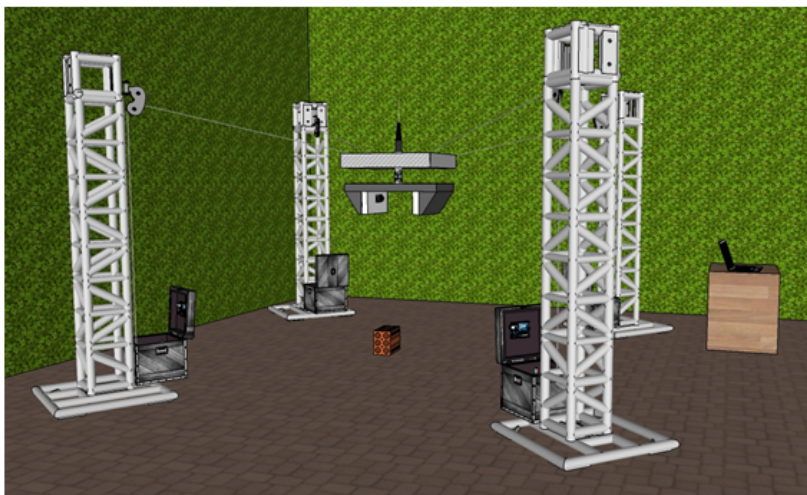


Figura 5.4: Vista lateral do sistema robótico completo

## 5.3 Especificação dos Equipamentos Utilizados na Implementação

Com a arquitetura do sistema bem estruturada, a próxima etapa a ser efetuada é a especificação dos equipamentos utilizados na implementação do sistema. Levando em consideração o que já foi apresentado nos capítulos anteriores, sabe-se que o primeiro dispositivo a ser definido é o cabo que será utilizado, posteriormente a arquitetura do guincho, seguido do sistema de sustentação superior do cabo, e por fim as placas de controlo e circuitos de acionamento dos atuadores.

As subsecções seguintes apresentam as características de cada dispositivo para satisfazer os requisitos do sistema.

### 5.3.1 Cabos

O primeiro componente a ser definido no projeto no robô é o cabo a ser utilizado para atuar o mesmo. Conforme referido na secção 5.1 (Requisitos do Sistema Robótico), foi estipulado uma carga útil de 5 kg, o que dá uma força gravitacional de 49 N. Além da carga útil, também deve ser levada em consideração a massa da plataforma, a ferramenta conectada a ela e os cabos: para estes componentes foram estipulados um acréscimo de 5 kg, levando a uma tensão máxima em cada cabo de aproximadamente 100 N.

A bitola do cabo interfere em diversos parâmetros do projeto, por exemplo, o comprimento do tambor, o passo da mesa, que por consequência influencia na relação de transmissão das engrenagens.

Para o dimensionamento dos cabos um dos fatores a ser considerado é o fator de segurança ( $n$ ), e este se deve ao tipo de trabalho onde será utilizado o cabo, para este projeto será adotado o que se utiliza em guinchos ( $n = 5$ ). O próximo valor a ser levado em consideração é a carga útil a ser suportada pelo cabo ( $C$ ), e para este caso  $C = 10$  kg. Utilizando a Equação 5.1 a carga de rutura ( $Cr$ ) pode ser expressa [35].

$$Cr = C.n \quad (5.1)$$

A partir da Equação 5.1 chega-se à carga de rutura de 50 kgf. A partir deste valor, deve-se recorrer à Tabela A.1 presente no Apêndice A e escolher o cabo que responda a estas condições.

Portanto opta-se pelo cabo de 1,6 mm de diâmetro, que suporta uma carga de rutura de 176 kgf.

### 5.3.2 Guincho

O próximo dispositivo a ser definido é a arquitetura do guincho. O tambor é o primeiro componente a ser definido, pois suas dimensões têm impacto na capacidade de cabos suportados pelo sistema, e também na capacidade do binário do guincho. A maior distância percorrida pela plataforma neste sistema será de 7,5 m (sendo o comprimento da maior diagonal), portanto o sistema deve ser capaz de suportar esta quantidade de cabos.

Utilizando um tambor com 62 mm de diâmetro, é possível enrolar 194 mm de cabo por revolução, sendo necessário cerca de 39 voltas para recolher ou dispensar



todo o cabo. Por se utilizar um cabo de 1,6 mm de diâmetro, e deixando uma folga de 0,4 mm entre os cabos, será necessário um tambor com 78 mm de comprimento.

Por se tratar de um sistema onde a quantidade de cabo dispensada tem suma importância em sua fiabilidade, foi inserido um dispositivo mecânico no guincho, que desempenha a função de guiar o cabo para a posição correta no tambor.

O dispositivo é composto por uma mesa linear acionada por fuso, acoplada à mesa está a peça por onde o cabo deve passar. O acionamento do fuso é feito por meio de uma correia, um lado conectada no tambor, e outro no fuso. Conforme o tambor gira, o fuso gira nas mesmas proporções de velocidade. Por se tratar de uma transmissão de potência direta, a inversão de direção do motor acarretará a inversão do fuso também. A Figura 5.5 ilustra o esquema mecânico do guincho.

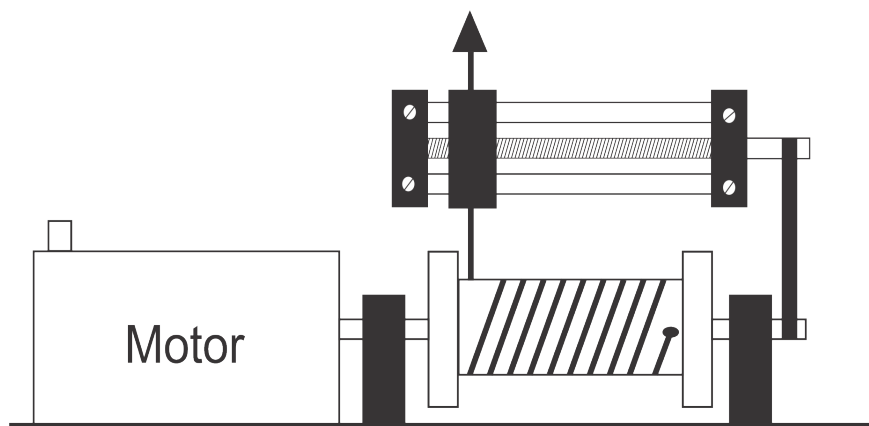


Figura 5.5: Vista lateral do esquema mecânico do guincho completo

Para a escolha da mesa linear dois parâmetros devem ser colocados: o passo e o comprimento do fuso. De acordo com a estrutura apresentada na Figura 5.5 o passo da mesa linear acionada por fuso deve ser igual ao valor do diâmetro do cabo mais o espaçamento, com isto é necessário um passo de 2 mm (para uma relação de transmissão de 1:1); outro valor a ser estipulado é o comprimento do fuso, e para este caso deve ser superior ao do tambor, sendo necessário um mínimo de 8 cm.

Por fim, para a escolha do motor, dois parâmetros devem ser apontados: binário e velocidade. Conforme apresentado na subsecção anterior, cada cabo deverá suportar aproximadamente 100 N. Utilizando a Equação 5.2 foi estipulado o binário mínimo necessário de 3,1 N.m para que o sistema responda corretamente.

$$T = F.r \quad , \quad (5.2)$$

onde  $T$  é o binário,  $F$  é a tensão no cabos e  $r$  é o raio do tambor.



Com o valor mínimo do binário necessário estabelecido (3,1 N.m), falta estabelecer a velocidade mínima de rotação do motor necessária para que o atuador final atinja a velocidade estabelecida nos requisitos do sistema.

Para que o atuador final atinja a velocidade de 0,25 m/s é necessário que o guincho libere ou recolha 25 cm de cabo a cada segundo. A cada revolução do tambor são dispensados ou recolhidos 19,4 cm de cabo, portanto é necessário um motor com velocidade mínima de rotação de 77 RPM.

Conforme visto na Figura 5.5 a fixação do tambor é feita por meio de duas chumaceiras. A fixação da mesa linear é feita por meio de parafusos que permitam regular a sua altura para que possa esticar a correia.

### 5.3.3 Sistema de Sustentação Superior do Cabo

O dispositivo de guia para o cabo na parte superior será composto por dois tarugos de ferro quadrado  $10 \times 10$  mm de 5 cm de comprimento, com um furo no meio de 5 mm de diâmetro por 5 mm de profundidade. Um eixo de 5 mm de diâmetro e 10 cm de comprimento se encaixa nos furos, fazendo a ligação entre os dois tarugos. O eixo é fixo, a parte móvel é composta por um tubo de ferro, o que está por fora do eixo, e soldada a ele duas chapas que fazem a sustentação da roldana. Entre o tubo e o eixo há um casquilho auto-lubrificante. Na Figura 5.6 pode ser visto os elementos que constituem este dispositivo.

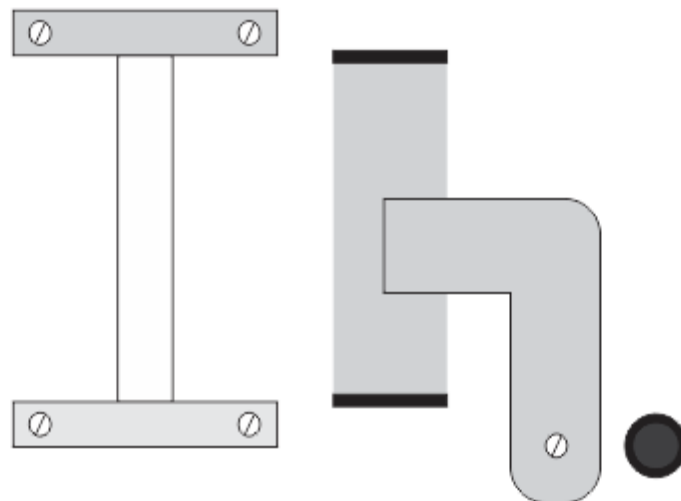


Figura 5.6: Componentes do sistema de sustentação superior do cabo

### 5.3.4 Controlo

Como referido na subsecção 5.2.2, o sistema de controlo é dividido em duas partes, o núcleo e os controladores individuais de cada motor. Por cada motor estar em um lugar distinto do outro, será necessário uma placa para cada motor, esta é a responsável por ler os dados enviados pelo computador (*kernel*), calcular a rotação equivalente do motor e enviar para o sistema de acionamento. Além desta será necessária uma placa para o controlo da orientação e estado da garra.

O *kernel* é implementado em qualquer computador com um sistema embebido, pois se trata de um software instalado na máquina, que calcula os comprimentos dos cabos para que a flange do robô se posicione na posição desejada, e depois os envie para as placas de controle junto aos motores.

## 5.4 Dispositivos Escolhidos para a Implementação do Sistema

Após a arquitetura do sistema ser definida, bem como as configurações de cada dispositivo utilizado, pode-se encontrar os dispositivos adequados para a implementação. Para chegar à decisão de quais seriam os componentes estudaram-se quatro soluções para o problema e fez-se uma tabela com as opções de componentes com suas características para cada uma delas; estas tabelas encontram-se no Apêndice B.

Tendo as quatro soluções, notou-se que a opção quatro apresenta a melhor relação custo/benefício.

Optou-se pelo servo motor Dynamixel MX-28T. Este motor oferece uma resolução para o controlo de posição de 12 bits para uma revolução, com isto apresenta uma precisão adequada. Apresenta uma dimensão de  $35,6 \times 50,6 \times 35,5$  mm, com uma massa de 77 g [36]. A velocidade máxima alcançada é de 67 RPM, fica abaixo da estipulada na subsecção 5.3.2 (77 RPM), porém analisando a relação custo/benefício a diferença entre as velocidades não mostrou grande impacto no resultado final. Suas especificações estão presentes na Tabela 5.1.

Para o acionamento do motor deve-se usar a placa de controlo OpenCM9.04, que exerce a função de receber os dados do *kernel* e comandar o motor. Esta placa possui interface para os periféricos de comunicação, e esta pode ser feita de três maneiras: Universal Serial Bus (USB), Bluetooth e ZigBee, dependendo apenas de qual periférico é conectado.

Para o posicionamento do cabo no tambor foi escolhida a mesa linear acionada por fuso drylin® SHTP-01-06-AWM, com um fuso de 6 mm de diâmetro, 2 mm de passo e 100 mm de comprimento, que apresenta as características necessárias para o sistema.

Tabela 5.1: Características do servo motor Dynamixel MX-28T [36]

Características	Valor
Tensão de Operação	11,1 V a 14,8 V
Binário	2,3 N.m a 3,1 N.m
Corrente	1,3 A a 1,7 A
Velocidade sem Carga	50 RPM a 67 RPM
Controle de Posição	Codificador 12 bits
Motor	Maxon
Redutora	193:1
Custo Médio	USD 219,90

## 5.5 Conclusão do Capítulo

Neste capítulo foi apresentada a arquitetura e a especificação detalhada do sistema, desde os requisitos, sua proposta inicial, até os componentes indicados para a implementação. Foram especificados os detalhes do projeto, bem como dimensões de componentes e esquemas mecânicos que facilitem o entendimento.

Tendo conhecimento de como se dá a cinemática do robô e qual é a sua arquitetura, o próximo passo para concluir o trabalho é implementar o sistema de simulação do robô, e este assunto será tratado no próximo capítulo.

## Capítulo 6

---

# Simulação de um Robô Paralelo Atuado a Cabos

---

*A simulação é muito importante para um sistema robótico, através dela sabe-se quais as competências do sistema, e pode-se verificar se é a melhor escolha para a solução do problema. Para os robôs a cabos não é diferente; levando isto em consideração, este capítulo apresenta a implementação do software de simulação do sistema robótico apresentado neste trabalho.*

### 6.1 Software de Simulação do Sistema

O *software* de simulação do sistema foi escrito em Java<sup>®</sup>, e foi escrito para que possa ser utilizado para fazer o controlo do robô. O simulador é composto de duas partes principais, a janela de interface gráfica e a janela de simulação do movimento.

A janela de interface gráfica é onde o utilizador deve introduzir os valores referentes às dimensões da estrutura e da plataforma do robô, a posição inicial e final do movimento realizado pelo atuador final, em qual tempo deve ocorrer o movimento, e se a garra deve estar aberta ou fechada e, por fim, se o movimento deve ser planeado para ser linear no espaço operacional, selecionando a instrução MoveL ou se o movimento deve ser planeado para ser linear no espaço das juntas selecionando a instrução MoveJ. A Figura 6.1 mostra a janela de interface gráfica desenvolvida para este robô.

A janela apresenta quatro botões: Padrão, Limpar, Carregar, Visualiza. O botão Padrão preenche os campos com as maiores dimensões permitidas para o robô, uma possibilidade de dimensões para a plataforma, e uma posição inicial e final para a garra. O botão Limpar retira os valores preenchidos nos campos

no interior do quadro Garra. O botão Carregar, quando pressionado, efetua todos os cálculos referentes ao movimento do robô, e ao final apresenta na tela os comprimentos iniciais e finais dos cabos, bem como as coordenadas  $x$ ,  $y$  e  $z$  das posições intermediárias da garra. Após o cálculo ter sido efetuado, pode-se pressionar o botão Visualizar para simular o movimento planejado, e uma nova janela será aberta na tela.

**Cable Robot**

**Estrutura**

Comprimento (X):

Largura (Y):

Altura (Z):

**Plataforma**

Comprimento (X):

Largura (Y):

Altura (Z):

Os valores introduzidos devem estar em centímetros

**Garra**

Posição da Garra:

Inicial Final

X:  Y:  Z:  X:  Y:  Z:

Situação da Garra:  Orientação:

☐ Aberta ☐ Fechada Tempo:

☐ MoveJ ☐ MoveL

**Comprimento dos Cabos**

	Inicial	Final
L1:	<input type="text"/>	<input type="text"/>
L2:	<input type="text"/>	<input type="text"/>
L3:	<input type="text"/>	<input type="text"/>
L4:	<input type="text"/>	<input type="text"/>

**Posição intermediária Garra**

X:

Y:

Z:

Figura 6.1: Janela de interface gráfica

Na janela de simulação, a trajetória é representada pelo traço em vermelho, e o *Tool Center Point* (TCP) é representado pelas linhas coloridas no centro da plataforma: a linha azul representando o eixo  $x$ , a verde o eixo  $y$  e a amarela o eixo  $z$ .

Para a simulação do sistema deve-se calcular as coordenadas  $x$ ,  $y$  e  $z$  em cada instante de tempo durante o movimento. Quando o movimento é linear no espaço operacional, ou seja, utiliza-se a instrução MoveL, o planeamento da trajetória calcula as coordenadas do atuador final em cada instante de tempo do movimento, e a partir delas são calculados os comprimentos dos cabos. A Figura 6.2 representa a janela de simulação para um movimento linear no espaço operacional.

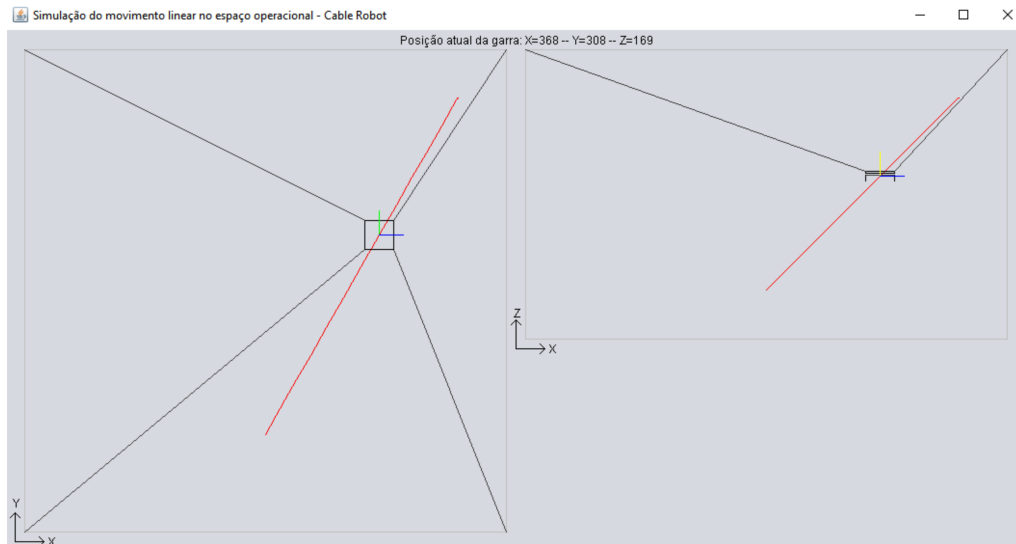


Figura 6.2: Janela de simulação do movimento linear no espaço operacional com a garra aberta

Quando o movimento é linear no espaço das juntas, ou seja, utiliza-se a instrução MoveJ, o planeamento da trajetória calcula o comprimento dos cabos para cada instante de tempo, e as coordenadas  $x$ ,  $y$  e  $z$  são obtidas através da cinemática direta para o espaço 3D, fazendo uso da Equação 4.11. Por se tratar de um sistema de equações não lineares, com quatro equações e três incógnitas, a solução deste sistema foi feita com o auxílio do *software* matemático Matlab<sup>®</sup>, onde o *software* de simulação envia o sistema, e o Matlab<sup>®</sup> retorna as coordenadas da posição correspondente àqueles comprimentos de cabos. A Figura 6.3 representa a janela de simulação para um movimento linear no espaço das juntas.

Após a implementação do sistema pode-se observar uma relação de erro entre as dimensões da estrutura e as da plataforma: quão maior as dimensões da plataforma e menor as dimensões da estrutura do robô, maior o erro, isto se dá pelo fato de que a Equação 4.11 apresenta uma aproximação da cinemática direta.

O código para geração do *software* de simulação do sistema estão presentes no Apêndice C.

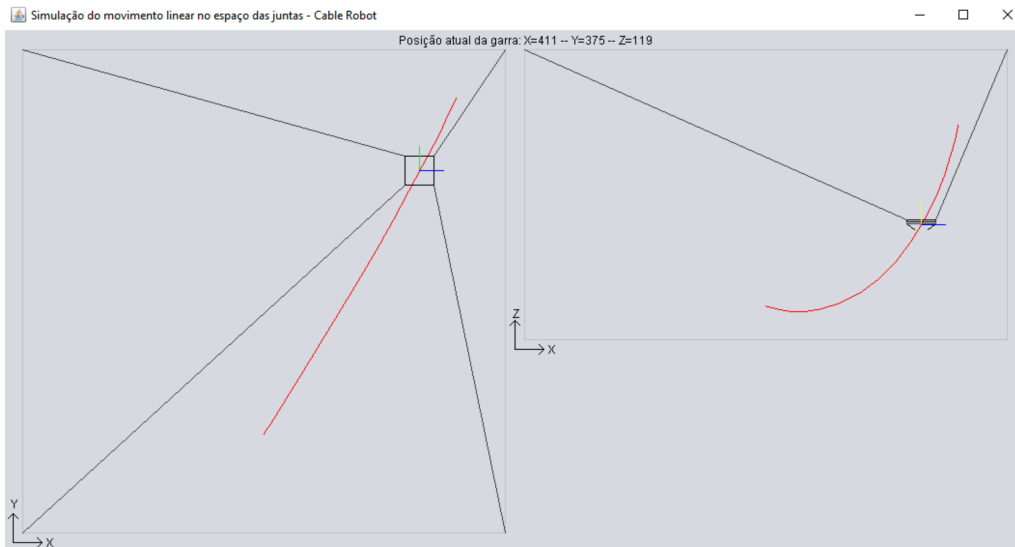


Figura 6.3: Janela de simulação do movimento linear no espaço das juntas com a garra fechada

## 6.2 Conclusão do Capítulo

Neste capítulo foi apresentado o último item faltante para a especificação e projeto de um robô a cabos: o *software* de simulação do sistema. O *software* foi implementado em Java<sup>®</sup>. Foram apresentadas as suas funcionalidades, bem como a janela de interface com o usuário e as janelas criadas para a simulação.

Com o *software* de simulação implementado, deve-se agora testar suas funcionalidades e avaliar os resultados obtidos, e este é o conteúdo apresentado no capítulo seguinte.

## Capítulo 7

---

# Testes e Resultados da Avaliação do Sistema

---

*A partir do software de simulação do sistema pode-se testar algumas configurações do robô proposto para verificar o seu funcionamento. O presente capítulo destina-se a apresentar os resultados obtidos com estas simulações.*

### 7.1 Simulação do Movimento

Para que o robô realize um movimento deve-se marcar a instrução MoveL ou MoveJ na janela de interface gráfica, estas instruções caracterizam o movimento a ser realizado.

A instrução MoveL deve ser utilizada quando se deseja que a plataforma realize o movimento em linha reta, já a instrução MoveJ deve ser utilizada para mover a plataforma rapidamente de um ponto para outro quando esse movimento não precisa estar em linha reta.

O primeiro teste apresentado utiliza uma estrutura para o robô com as dimensões  $5 \times 5 \times 3$  m e uma plataforma com as dimensões de  $0,3 \times 0,3 \times 0,1$  m. O teste baseou-se em mover a plataforma, da posição inicial com coordenadas (1; 1,4; 0,5) até a posição final com coordenadas (4,5; 4,5; 2,5) em um intervalo de tempo de 4 segundos. Para verificar a funcionalidade do simulador utilizaram-se as duas instruções relativas ao movimento, MoveL e MoveJ, e os resultados obtidos com estas simulações estão representados nas Figuras 7.1 e 7.2.

Observando os resultados obtidos com a simulação pôde-se notar que quando utilizada a instrução MoveL o resultado alcançado é o esperado, ou seja, a plataforma parte da posição inicial e em linha reta realiza sua trajetória até a posição



final conforme indicadas na janela de interface gráfica. Já quando é utilizada a instrução MoveJ a simulação apresenta um resultado aproximado, pois as posições iniciais e finais não correspondem exatamente às especificadas.

Para obter resultados mais claros em relação à instrução MoveJ realizaram-se três simulações com tamanhos diferentes para a plataforma e uma estrutura com dimensões de  $5 \times 5 \times 3$  m. No primeiro caso utilizou uma plataforma com dimensões de  $0,5 \times 0,5 \times 0,3$  m, o segundo caso utilizou uma plataforma com as dimensões de  $0,1 \times 0,1 \times 0,1$  m e, por fim, uma com apenas um ponto para conexão dos cabos. O movimento realizado pelas simulações tem posição inicial com coordenadas (1,4; 1; 1) e posição final com coordenadas (4,5; 4,5; 2,5), e foi realizado em um intervalo de tempo de 4 segundos. Os resultados obtidos com estas simulações estão representados nas Figuras 7.3, 7.4 e 7.5.

## 7.2 Erro

A partir das simulações pôde-se observar que o quão maior for a distância entre o TCP e o ponto de ancoragem do cabo maior o erro de posicionamento. Para expressar o valor do erro do erro de posicionamento foram retiradas as coordenadas de cinco posições intermédias entre o ponto de partida e o ponto de chegada, este erro é proveniente da aproximação utilizada na equação da cinemática direta-Equação 4.11; estas coordenadas estão organizadas na Tabela 7.1 para melhor visualização. Foram adotados como valores de referência as coordenadas retiradas da simulação onde se utilizou apenas um ponto para ancoragem dos cabos.

Tabela 7.1: Coordenadas retiradas das simulações

Percurso realizado	Referência			1º caso			2º caso		
	$x$	$y$	$z$	$x$	$y$	$z$	$x$	$y$	$z$
0%	140	100	100	141	102	61	139	99	88
25%	176	140	77	178	142	43	176	140	67
50%	307	287	75	308	288	46	307	287	67
75%	422	418	178	423	419	142	423	419	168
100%	450	450	250	450	450	194	450	450	232

A partir dos dados das coordenadas encontradas na Tabela 7.1 pode-se utilizar a Equação 7.1 e expressar os valores referentes ao erro de posicionamento, e estes valores estão presentes na Tabela 7.2.

$$\Delta i = i_0 - i, \quad i = x, y, z. \quad (7.1)$$

Analisando os valores obtidos a partir dos cálculos do erro pode-se afirmar que as dimensões da plataforma têm influência significativa na precisão da posição da

Tabela 7.2: Erro absoluto

Percurso realizado	Erro absoluto [cm]					
	1º caso			2º caso		
	$x$	$y$	$z$	$x$	$y$	$z$
0%	1	2	39	0	1	12
25%	2	2	34	0	0	10
50%	1	1	29	0	0	8
75%	1	1	36	1	1	10
100%	0	0	56	0	0	18
Mínimo	0	0	29	0	0	8
Máximo	2	0	56	1	1	18
Média	1	1,2	38,8	0,2	0,4	11,6

plataforma. Para solucionar este problema deve-se melhorar a metodologia utilizada para encontrar o comprimento do raio das esferas utilizadas na cinemática direta.

### 7.3 Conclusão do Capítulo

Neste capítulo foram apresentados os resultados obtidos a partir da simulação de alguns movimentos, e a partir da análise destes pôde-se observar algumas divergências entre o resultado obtido e o esperado. Foi feito o estudo do erro, e notou-se que o sistema apresenta maior exatidão quando se utiliza uma plataforma com pequenas dimensões.

O capítulo seguinte faz a conclusão de todo o trabalho, fazendo um balanço entre o que foi feito e o que pode ser feito em um trabalho futuro.

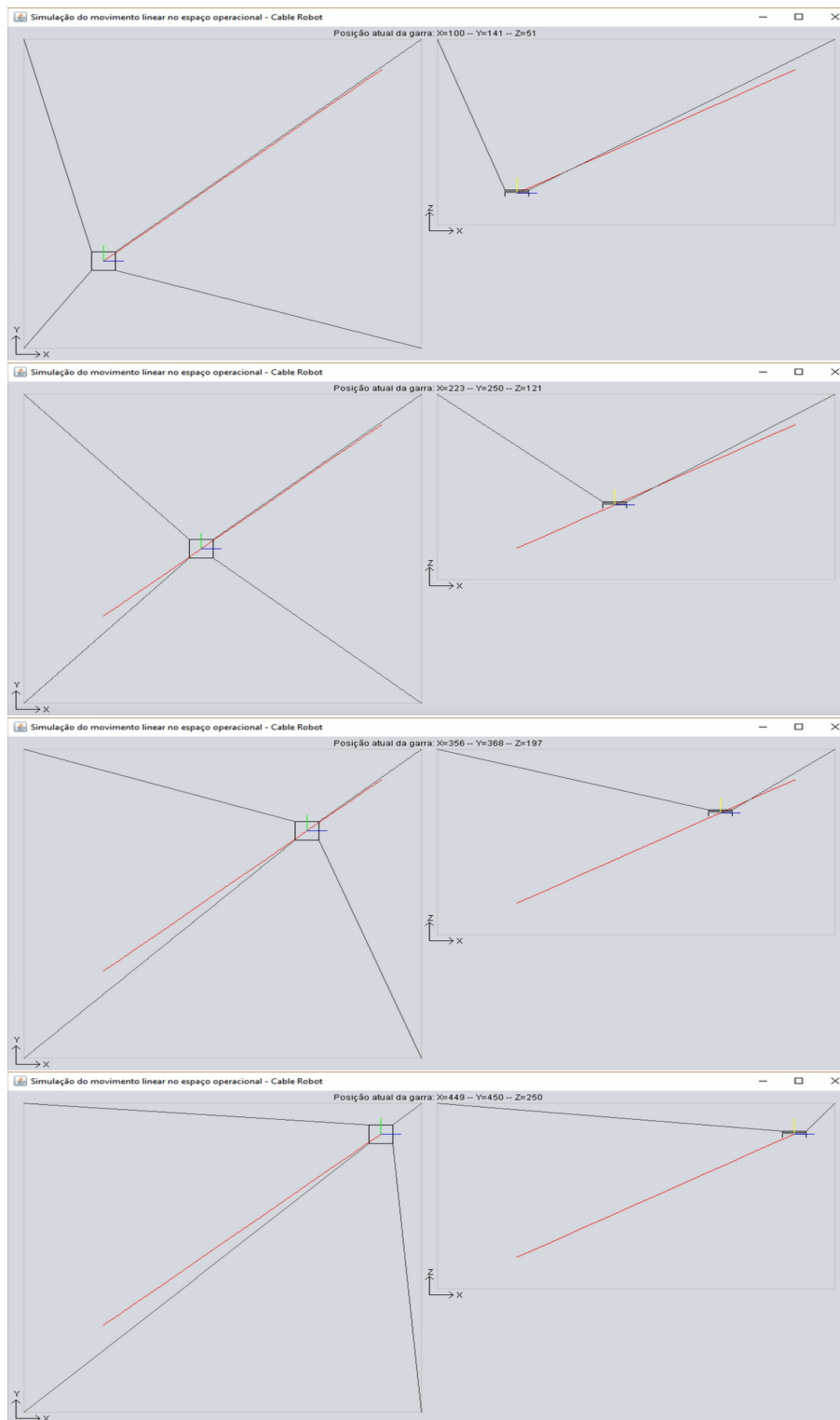


Figura 7.1: Sequência do movimento nos planos  $x-y$  (esquerda) e  $x-z$  (direita) com o planeamento da trajetória linear no espaço operacional

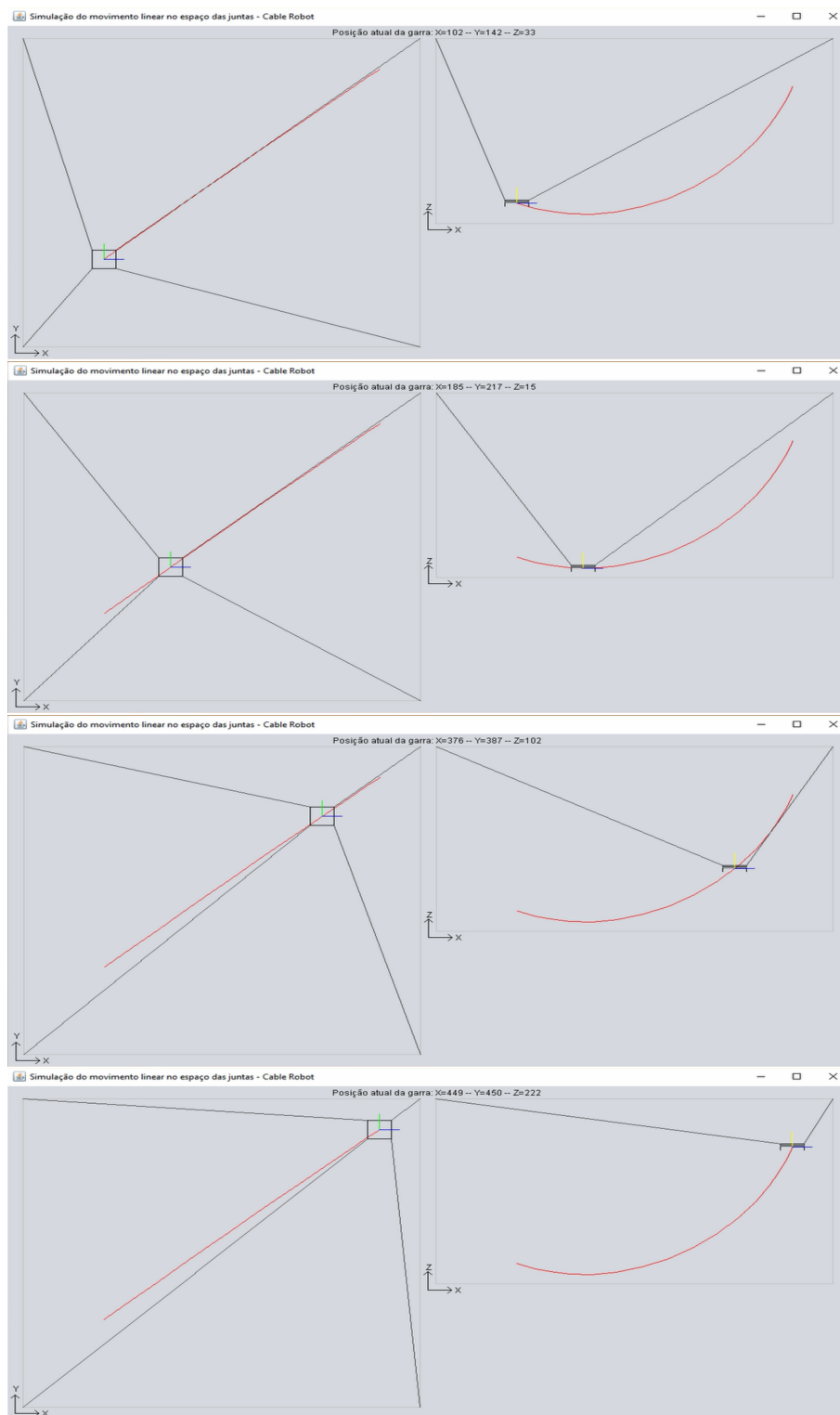


Figura 7.2: Sequência do movimento nos planos  $x - y$  (esquerda) e  $x - z$  (direita) com o planeamento da trajetória linear no espaço das juntas

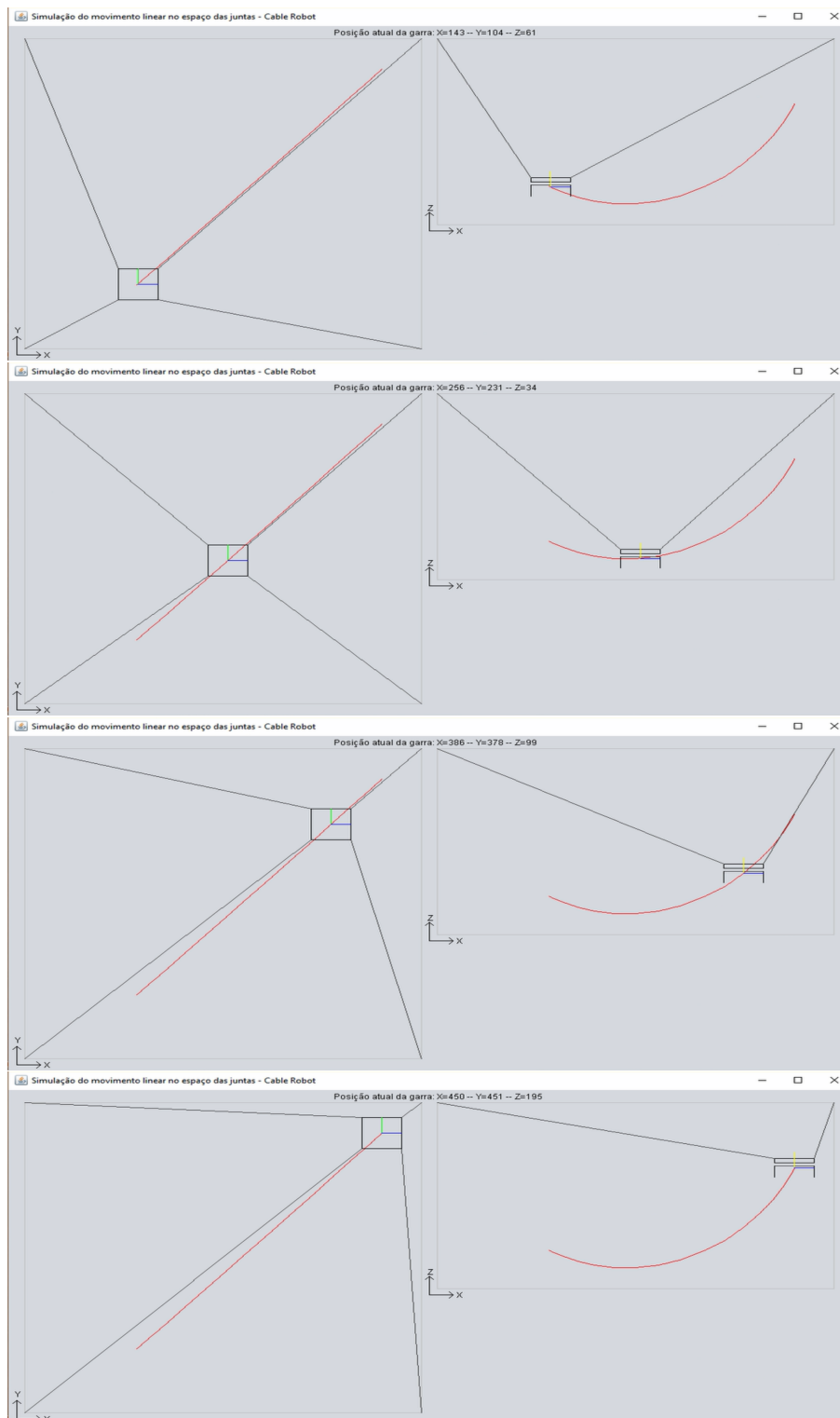


Figura 7.3: Aspeto geral da simulação utilizando a plataforma com dimensões  $0,5 \times 0,5 \times 0,3$  m

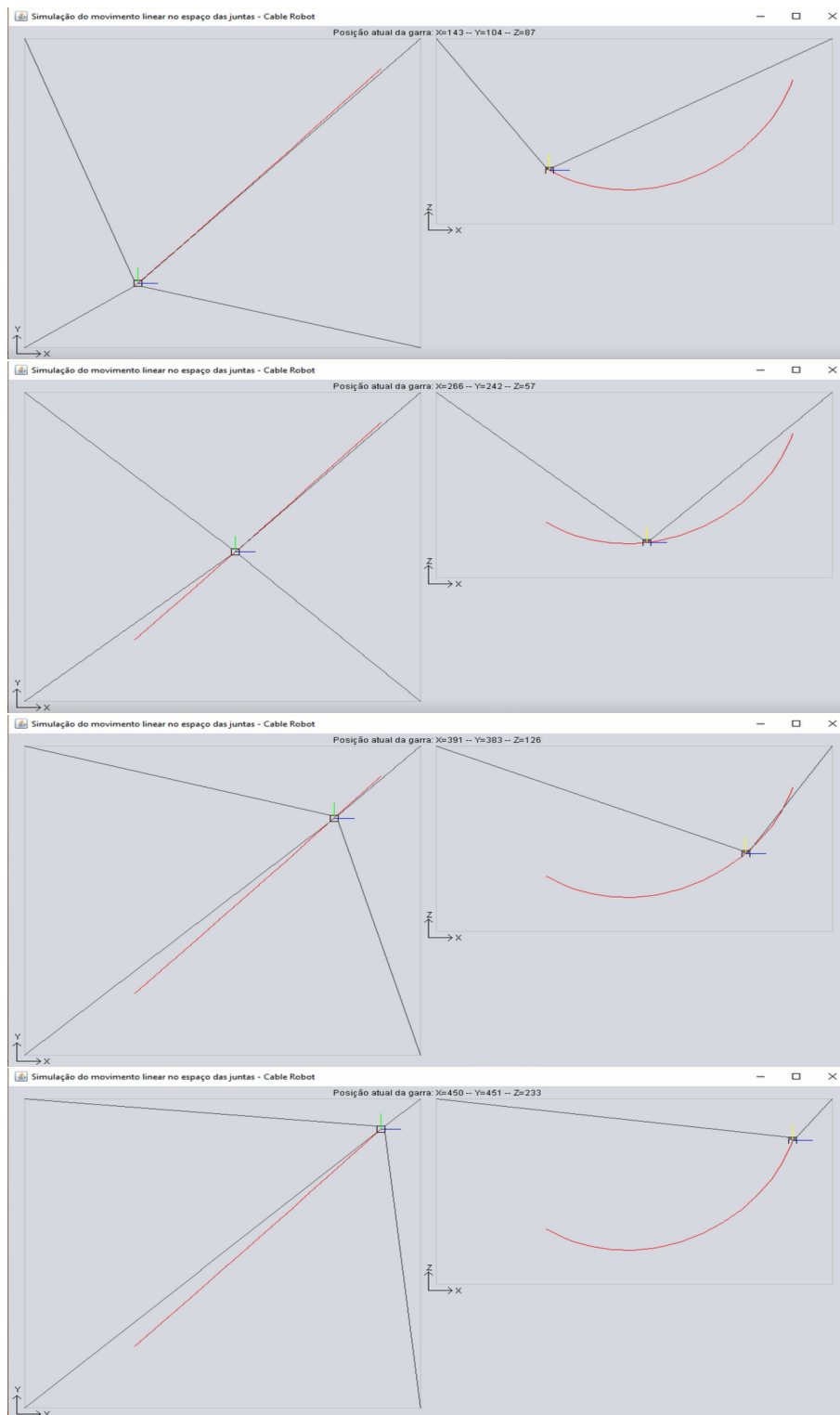


Figura 7.4: Aspeto geral da simulação utilizando a plataforma com dimensões  $0,1 \times 0,1 \times 0,1$  m

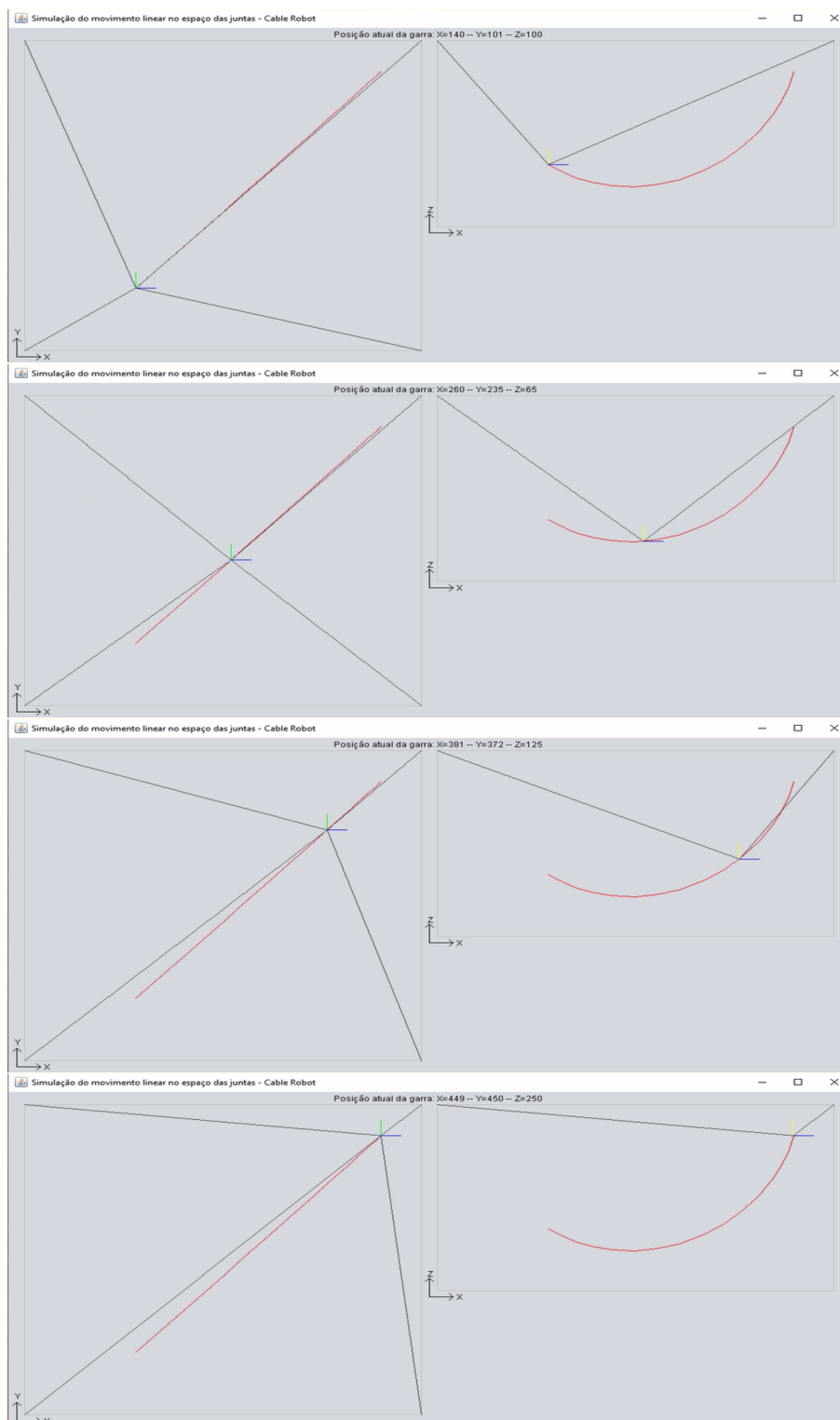


Figura 7.5: Aspecto geral da simulação utilizando apenas um ponto para conexão dos cabos.

## Capítulo 8

---

# Conclusão e Desenvolvimentos Futuros

---

*Neste capítulo realiza-se o balanço do trabalho realizado face aos propósitos iniciais e apontam-se os desenvolvimentos futuros da aplicação.*

### 8.1 Balanço da Obra

Durante os sete meses de trabalho foram alcançadas as seguintes metas:

- Estudo dos robôs atuados por cabos;
- Estudo dos sensores e atuadores;
- Desenvolvimento da arquitetura mecânica;
- Estudo e implementação da cinemática;
- Desenvolvimento do planeamento da trajetória;
- Implementação do *software* de simulação;
- Escolha dos dispositivos eletrónicos e mecânicos para implementação.

Por uma questão temporal não foi possível a implementação do sistema robótico por completo, uma vez que o processo de compra dos dispositivos necessários levaria algum tempo.



## 8.2 Conclusão

A família de robôs paralelos atuados por cabos tem muito a oferecer e a ser estudada, por se tratar de um conceito relativamente novo, ainda há muitos parâmetros a serem aprimorados.

Neste trabalho foi proposto um robô atuado por quatro cabos, adequado para a finalidade pretendida e com um custo reduzido face aos robôs que utilizam uma quantidade maior de cabos. Foi proposto que a comunicação entre os guinchos e o *kernel* do sistema de controlo seja feita via rádio, facilitando a reconfigurabilidade do robô.

Outro fator apresentado neste trabalho é a possibilidade de um volume de trabalho variável, podendo utilizar somente o necessário para a aplicação desejada, o que o torna mais versátil.

Foi desenvolvido um *software* de simulação para o robô proposto levando em consideração os requisitos básicos e a arquitetura proposta: a partir dele é possível visualizar o movimento desejado nos planos  $x - y$  e  $x - z$ .

No processo de desenvolvimento do sistema foi possível a aquisição de conhecimentos das mais diversas áreas, como: geometria, cinemática, motores elétricos, mecânica dos cabos e programação, dentre outras.

## 8.3 Desenvolvimentos Futuros

Visto que a presente dissertação focou-se em planejar a arquitetura, o circuito de controlo e a simulação do mesmo, pode-se em um próximo trabalho implementar a arquitetura proposta de forma a analisar as velocidades e acelerações alcançadas.

Pode-se fazer uma melhoria na interface entre usuário e robô, para que o sistema seja capaz de receber um arquivo do tipo CAD, interpretar e enviar as coordenadas ao robô para que ele execute o projeto.

Outra melhoria interessante para o sistema é a simulação acontecer no espaço 3D, facilitando a visualização do movimento para o usuário.

---

# Bibliografia

---

- [1] Contour Crafting, HOW DOES IT WORK?. Disponível em: <<http://www.contourcrafting.org/how-does-it-work/>>. Acesso em 6 de março de 2017. [citado na p. 1]
- [2] Contour Crafting, CONTOUR CRAFTING. Disponível em: <<http://www.contourcrafting.org/>>. Acesso em 6 de março de 2017. [citado na p. 1]
- [3] Challacombe, BJ, Khan, MS, Murphy, D. et al. World J Urol (2006) 24: 120. [citado na p. 5, 6]
- [4] Rosheim, Mark E. Robot Evolution The development of anthrobotics. [citado na p. 5]
- [5] Roberts, Adam. "Introduction", to RUR & War with the Newts. London, Gollancz, 2011 [citado na p. 5]
- [6] Clarke R. Asimov's laws for robotics: Implications for Information Technology. Computer 1993; January: 313–320. 1994; Part 1–2 December: 53–61. [citado na p. 5]
- [7] Robotics, UNIMATE//The First Industrial Robot. Disponível em: <<http://www.robotics.org/joseph-engelberger/unimate.cfm>>. Acesso em 23 de fevereiro de 2017. [citado na p. 6]
- [8] Assembly robot US Pat. 4,341,502. Disponível em: <<https://docs.google.com/viewer?url=patentimages.storage.googleapis.com/pdfs/US4341502.pdf>>. Acesso em 23 de fevereiro de 2017. [citado na p. 6]
- [9] S. B. Niku, Introdução à Robótica, Rio de Janeiro: ed. LTC - Livros Técnicos e Científicos Editora Ltda, 2013 [citado na p. 6]
- [10] Inc., Robotics. Disponível em: <<https://www.inc.com/encyclopedia/robotics.html>>. Acesso em 25 de fevereiro de 2017. [citado na p. 6]

- [11] Pott, Andreas; Mütterich, Hendrick; Kraus, Werner et al. IPAnema: A family of Cable-Driven Parallel Robots for Industrial Applications. 2013. In book: Cable-Driven Parallel Robots, pp.119-134. [citado na p. xiii, 6, 9, 10, 11, 18]
- [12] Bruckmann, Tobias; Pott, Andreas. Cable-Driven Parallel Robots: Mechanisms and Machine Science. 12. Berlin: Springer-Verlag, 2013. [citado na p. 6, 7]
- [13] NIST, RoboCrane® - Fabricação em Grande Escala com Controle de Cabos. Disponível em: <<https://www.nist.gov/intelligent-systems-division/robocrane%C2%AE-large-scale-manufacturing-using-cable-control>>. Acesso em 1 de março de 2017. [citado na p. 7]
- [14] Bostelman, Roger V; Albus, James S (1993). STABILITY OF AN UNDERWATER WORK PLATFORM SUSPENDED FROM AN UNSTABLE REFERENCE. Disponível em:<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=326114>> Acesso em 21 de março de 2017. [citado na p. 7]
- [15] Journal of Research of the National Institute of Standards and Technology, The NIST RoboCrane, Volume 97, Number 3, May-June 1992 [citado na p. xiii, 7, 8]
- [16] NIST, Air Craft Maintenance. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/air-craft-maintenance>>. Acesso em 1 de março de 2017. [citado na p. 8]
- [17] NIST, Air Transportable Expeditionary Crane. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/atec1>>. Acesso em 1 de março de 2017. [citado na p. 9]
- [18] NIST, Bridge Construction. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/bridge-construction>>. Acesso em 1 de março de 2017. [citado na p. 9]
- [19] NIST, Lunar Rover. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/lunar-rover>>. Acesso em 1 de março de 2017. [citado na p. xiii, 9]
- [20] NIST, Machining. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/machining>>. Acesso em 1 de março de 2017. [citado na p. 9]
- [21] NIST, Off-Shore Cargo Handling. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/shore-cargo-handling>>. Acesso em 1 de março de 2017. [citado na p. 9]

- [22] NIST, Ship Building. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/ship-building>>. Acesso em 1 de março de 2017. [citado na p. 9]
- [23] NIST, Undersea Configurations. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/undersea-configurations>>. Acesso em 1 de março de 2017. [citado na p. xiii, 9]
- [24] NIST, Oil-Well Fire Fighting. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/oil-well-fire-fighting>>. Acesso em 1 de março de 2017. [citado na p. 9]
- [25] Auffray, Valérie. Vers la manipulation précise de grandes pièces dans de très grands espaces de travail. Disponível em: <[http://www.cnrs.fr/mi/IMG/pdf/cable\\_tecnalia\\_cnrs.pdf](http://www.cnrs.fr/mi/IMG/pdf/cable_tecnalia_cnrs.pdf)> Acesso em 22 de março de 2017. [citado na p. xiii, 11, 13, 14]
- [26] Izard, Jean-Baptiste; Gouttefarde, Marc; Michelin, Micaël et al. A Reconfigurable Robot for Cable-Driven Parallel Robotic and Industrial Scenario Proving. 2013. In book: Cable-Driven Parallel Robots, pp.135-148 [citado na p. xiii, 11, 12, 13]
- [27] Interempresas, Validación del prototipo de robot de cables CoGiRo para tareas de manipulación y ensamblaje de piezas complejas de alto valor económico. Disponível em: <<http://www.interempresas.net/MetalMecanica/Articulos/144648-Validacion-prototipo-robot-cables-CoGiRo-tareas-manipulacion-ensamblaje-piezas-complejas.html>>. Acesso em 23 de março de 2017. [citado na p. xiii, 14]
- [28] CableBOT, summary. Disponível em: <<http://www.cablebot.eu/en/>>. Acesso em 24 de fevereiro de 2017. [citado na p. xiii, 15]
- [29] Fitzgerald, A.E. “Máquinas elétricas”, 6ª Ed., São Paulo: McGraw-Hill, 2006. Pag 418 a 422. [citado na p. xiii, 19]
- [30] Andrade, Alan Sulato de. ELEMENTOS ORGÂNICOS DE MÁQUINAS II. Disponível em:<<http://www.madeira.ufpr.br/disciplinasalan/AT102-Aula04.pdf>> Acesso em 27 de março de 2017. [citado na p. 20]
- [31] Matos, N. M. R. Análise do Funcionamento de um Servomotor de Corrente Alternada com Ímãs Permanentes. 2012.Dissertação - Faculdade de Engenharia da Universidade do Porto. [citado na p. 20]
- [32] Thomazini, Daniel; Albuquerque, Pedro Urbano Braga de. Sensores industriais: fundamentos e aplicações. 8ed. SP: Érica, 2011. [citado na p. 21]

- [33] FRADEN, JACOB. HANDBOOK OF MODERN SENSORS: PHYSICS, DESIGNS, and APPLICATIONS. 3<sup>o</sup> edition. San Diego, California. Springer-Verlag, 2004. Pp. 323-336. [citado na p. xiii, 22]
- [34] M.W. Spong and M. Vidyasagar, Robot Dynamics and Control. New York: Wiley, 1989 [citado na p. 34]
- [35] Raul. Cabos de aço: usos e aplicações. Rio de Janeiro: [s. n.], c1986. [citado na p. xv, 46, 69]
- [36] Datasheet, ROBOTIS. Disponível em: <[http://support.robotis.com/en/product/actuator/dynamixel/mx\\_series/mx-28at\\_ar.htm](http://support.robotis.com/en/product/actuator/dynamixel/mx_series/mx-28at_ar.htm)>. Acesso em 5 de junho de 2017. [citado na p. xv, 49, 50]

## Apêndice A

# Parâmetros do Cabo de Aço

Tabela A.1: Cargas de ruptura e massa linear [35]

DIÂMETRO		MASSA POR METRO LINEAR (Kg / m)				Cargas de ruptura - Resistência 180 / 200 Kg/mm2 (I.P.S.) em Kgf			
POL	MM	6x7		6x19 / 6x25 / 6x37		6x7		6X19 / 6X25 / 6X37	
								AA / AACI	AF / AFA
1/16"	1,6	0,012	0,013	-	-	176	-	-	-
5/94"	2	0,014	0,015	-	-	236	259	-	-
3/32"	2,4	0,019	0,022	-	-	340	365	-	-
1/8"	3,2	0,034	0,037	0,039	0,043	600	646	-	-
5/32"	4	0,055	0,060	-	-	1100	1166	-	-
3/16"	4,8	0,078	0,086	0,088	0,096	1350	1449	1400	1500
1/4"	6,4	0,140	0,154	0,156	0,171	2390	2571	2480	2660
5/16"	8	0,220	0,205	0,244	0,267	3720	4153	3860	4150
3/8"	9,5	0,310	0,320	0,351	0,382	5320	5714	5530	5940
7/16"	11,5	0,430	-	0,476	0,528	7190	7735	7500	8060
1/2"	13	0,560	0,585	0,625	0,684	9340	10051	9710	10410
9/16"	14,5	0,710	-	0,788	0,878	11800	12755	12200	13110
5/8"	16	0,880	0,960	0,982	1,071	14400	15510	15100	16230
3/4"	19	1,250	-	1,413	1,548	20600	22143	21600	23220
7/8"	22	-	-	1,919	2,113	-	-	29200	31390
1"	26	-	-	2,500	2,753	-	-	37900	40740







## Apêndice B

# Dispositivos para Implementação

### B.1 Opção 1



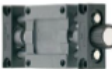

Tabela B.1: Primeira solução encontrada

Componente	Nome	Link	Foto	Quantidade	Preço	Total
Servo motor	Dynamixel MX-64R	<a href="http://www.trossenrobotics.com/p/MX-64R-dynamixel-robot-actuator.aspx">http://www.trossenrobotics.com/p/MX-64R-dynamixel-robot-actuator.aspx</a>		01	\$305.90	\$305.90
Chumaceiras-diâmetro interno 5mm	Chumaceira com flange igubal® - KSTM-	<a href="http://www.igus.pt/wpck/2486/igubal_Stehlager_KST">http://www.igus.pt/wpck/2486/igubal_Stehlager_KST</a>		02	4,51 euros	9,02 euros
Tambor	DESCRIÇÃO - Tarugo de alumínio redondo de 150 mm e 35 mm, torneado 2 mm no diâmetro e deixando 1.5 mm em cada ponta no comprimento para fazer borda lateral e, com isso fazer a cave para o cabo do cabo. Um tambor com diâmetro de 146 mm e 35 mm de comprimento.			01	-	-
Mesa linear	SHTP-01-06-AWM	<a href="http://www.igus.pt/wpck/3373/DryLin_SHTP_Min_i?C=PT&amp;L=pt">http://www.igus.pt/wpck/3373/DryLin_SHTP_Min_i?C=PT&amp;L=pt</a>		01	78,89 euros	78,89 euros
Eixo	5mm	-	-	10 cm	-	-
Buchas de fixação para o eixo	Pololu universal aluminum mounting hub for 5mm shaft, m3 holes	<a href="http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes">http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes</a>		Em cada pack vem duas, portanto precisamos de dois packs -02	\$7,00	\$14,00







## B.2 Opção 2

Tabela B.2: Segunda solução encontrada

Componente	Nome	Link	Foto	Quantidade	Preço	Total
Servo motor	DYNAMIXEL RX-24F HN07-N101 Type	<a href="http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes">http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes</a>		01	\$139,90	\$139,90
Chumaceiras-diâmetro interno 5mm	Chumaceira com flange igubal® - KSTM-	<a href="http://www.igus.pt/wpck/2486/igubal_Stehlager_KST">http://www.igus.pt/wpck/2486/igubal_Stehlager_KST</a>		02	4,51 euros	9,02 euros
Tambor	DESCRIÇÃO- Tarugo de alumínio redondo de 50mm e 10 cm, ou tarugo de 55 mm torneado 2,5mm fazendo a cave do cabo.			01	-	-
Mesa linear	SHTP-01-06-AWM	<a href="http://www.igus.pt/wpck/3373/DryLin_SHTP_Min_i?C=PT&amp;L=pt">http://www.igus.pt/wpck/3373/DryLin_SHTP_Min_i?C=PT&amp;L=pt</a>		01	78,89 euros	78,89 euros
Eixo	5mm	-	-	10 cm	-	-
Buchas de fixação para o eixo	Pololu universal aluminum mounting hub for 5mm shaft, m3 holes	<a href="http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes">http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes</a>		Em cada pack vem duas, portanto precisamos de dois packs -02	\$7,00	\$14,00





## B.3 Opção 3

Tabela B.3: Terceira solução encontrada

Componente	Nome	Link	Foto	Quantidade	Preço	Total
Servo motor	Mechaduino 0.2 Servo	<a href="http://tropical-labs.com/index.php/product/mechaduino-0-2-servo">http://tropical-labs.com/index.php/product/mechaduino-0-2-servo</a>		01	\$ 65,00	\$ 65,00
Chumaceiras-diâmetro interno 5mm	Chumaceira com flange igubal® - KSTM-	<a href="http://www.igu.s.pt/wpck/2486/igubal_Stehlager_KST">http://www.igu.s.pt/wpck/2486/igubal_Stehlager_KST</a>		02	4,51 euros	9,02 euros
Tambor	<p>DESCRIÇÃO -</p> <p>Tarugo de alumínio redondo de 12 mm e 270 mm, torneado 1.5 mm no diâmetro e deixando 2.5 mm em cada ponta no comprimento para fazer borda lateral e, com isso fazer a cave para o cabo do cabo. Um tambor com diâmetro de 9 cm e 265 mm de comprimento.</p>			01	-	-
Mesa linear	SHTP-01-06-AWM	<a href="http://www.igus.pt/wpck/3373/DryLin_SHTP_Mini?C=PT&amp;L=pt">http://www.igus.pt/wpck/3373/DryLin_SHTP_Mini?C=PT&amp;L=pt</a>		01	78,89 euros	78,89 euros
Eixo	5mm	-	-	10 cm	-	-
Buchas de fixação para o eixo	Pololu universal aluminum mounting hub for 5mm shaft, m3 holes	<a href="http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes">http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes</a>		Em cada pack vem duas, portanto precisamos de dois packs -02	\$7,00	\$14,00

## B.4 Opção 4

Tabela B.4: Quarta solução encontrada

Componente	Nome	Link	Foto	Quantidade	Preço	Total
Servo motor	Dynamixel MX-28T 3.1N.m/ 67RPM	<a href="http://www.robotshop.com/en/dynamixel-mx-28t-serial-servo-ttl.html">http://www.robotshop.com/en/dynamixel-mx-28t-serial-servo-ttl.html</a>		01	\$219,90	\$219,90
Chumaceiras-diâmetro interno 5mm	Chumaceira com flange igubal® - KSTM-	<a href="http://www.igus.pt/wpck/2486/igubal_Stehlager_KST">http://www.igus.pt/wpck/2486/igubal_Stehlager_KST</a>		02	4,51 euros	9,02 euros
Tambor	DESCRIÇÃO - Tarugo de alumínio redondo de 65 mm e 80 mm, torneado 1,5 mm no diâmetro e deixando 1 mm em cada ponta no comprimento para fazer borda lateral e, com isso fazer a cave para o cabo do cabo. Um tambor com diâmetro de 62 mm e 78 mm de comprimento.			01	-	-
Mesa linear	SHTP-01-06-AWM	<a href="http://www.igus.pt/wpck/3373/DryLin_SHTP_Min?C=PT&amp;L=pt">http://www.igus.pt/wpck/3373/DryLin_SHTP_Min?C=PT&amp;L=pt</a>		01	78,89 euros	78,89 euros
Eixo	5mm	-	-	10 cm	-	-
Buchas de fixação para o eixo	Pololu universal aluminum mounting hub for 5mm shaft, m3 holes	<a href="http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes">http://www.robotpark.com/Pololu-Universal-Aluminum-Mounting-Hub-for-5mm-Shaft-M3-Holes</a>		Em cada pack vem duas, portanto precisamos de dois packs -02	\$7,00	\$14,00

## Apêndice C

---

# Códigos em Java<sup>®</sup> para Geração do Software de Simulação do Sistema

---

### C.1 Código do Cálculo da Cinemática e Planeamento das Trajetórias

```
1  Indice=0;
2  DecimalFormat df = new DecimalFormat("0.00");
3  Xi = Integer.parseInt(jTextField1.getText());
4  Yi = Integer.parseInt(jTextField2.getText());
5  Zi = Integer.parseInt(jTextField3.getText());
6  Xf = Integer.parseInt(jTextField15.getText());
7  Yf = Integer.parseInt(jTextField16.getText());
8  Zf = Integer.parseInt(jTextField17.getText());
9  PlatX = Integer.parseInt(jTextField8.getText());
10 PlatY = Integer.parseInt(jTextField9.getText());
11 PlatZ = Integer.parseInt(jTextField10.getText());
12 EstX = Integer.parseInt(jTextField11.getText());
13 EstY = Integer.parseInt(jTextField12.getText());
14 EstZ = Integer.parseInt(jTextField13.getText());
15 Tf = Float.parseFloat(jTextField22.getText());
16 orientacao = Integer.parseInt(jTextField14.getText());
17
18 if(Omove==true){
19     tipo="linear no espaco operacional";
20
21     //Calculando os coeficientes do polinomio
```

```

22 //Para X
23 A0x = Xi;
24 A3x=(float) ((20*(Xf-Xi))/(2*Math.pow(Tf,3)));
25 A4x=(float) ((30*(Xi-Xf))/(2*Math.pow(Tf,4)));
26 A5x=(float) ((12*(Xf-Xi))/(2*Math.pow(Tf,5)));
27
28 //Para Y
29 A0y = Yi;
30 A3y=(float) ((20*(Yf-Yi))/(2*Math.pow(Tf,3)));
31 A4y=(float) ((30*(Yi-Yf))/(2*Math.pow(Tf,4)));
32 A5y=(float) ((12*(Yf-Yi))/(2*Math.pow(Tf,5)));
33 //Para Z
34 A0z = Zi;
35 A3z=(float) ((20*(Zf-Zi))/(2*Math.pow(Tf,3)));
36 A4z=(float) ((30*(Zi-Zf))/(2*Math.pow(Tf,4)));
37 A5z=(float) ((12*(Zf-Zi))/(2*Math.pow(Tf,5)));
38 for ( float i = 0; i<Tf; i=(float) (i+Tf/100)){
39     Xt [Indice] = (float) ((float) A0x+(A3x*Math.pow(i,3))
40         +(A4x*Math.pow(i,4))+(A5x*Math.pow(i,5)));
41     Yt [Indice] = (float) ((float) A0y+(A3y*Math.pow(i,3))
42         +(A4y*Math.pow(i,4))+(A5y*Math.pow(i,5)));
43     Zt [Indice] = (float) ((float) A0z+(A3z*Math.pow(i,3))
44         +(A4z*Math.pow(i,4))+(A5z*Math.pow(i,5)))
45     R11 [Indice] = (float) Math.sqrt(Math.pow((Xt[Indice]
46         ]-(PlatX/2)-0),2)+ Math.pow((Yt[Indice]-(PlatY/2)
47         -0),2)+ Math.pow((Zt[Indice]-(PlatZ/2)-EstZ),2));
48     R12 [Indice] = (float) Math.sqrt(Math.pow((Xt[Indice]
49         ]+(PlatX/2)- EstX),2)+Math.pow((Yt[Indice]-(PlatY
50         /2)- 0),2)+Math.pow((Zt[Indice]-(PlatZ/2)-EstZ),2)
51         );
52     R13 [Indice] = (float) Math.sqrt(Math.pow((Xt[Indice]
53         ]+(PlatX/2)- EstX),2)+Math.pow((Yt[Indice]+(PlatY
54         /2)- EstY),2)+Math.pow((Zt[Indice]-(PlatZ/2)-EstZ)
55         ,2));
56     R14 [Indice] = (float) Math.sqrt(Math.pow((Xt[Indice]
57         ]-(PlatX/2)-0),2)+Math.pow((Yt[Indice]+(PlatY/2)-
58         EstY),2)+Math.pow((Zt[Indice]-(PlatZ/2)-EstZ),2));
59     Indice++;
60 }
61 for ( int j = 0; j<Indice; j++){
62     jTextField23.setText(String.valueOf(df.format(Xt[j])))
63     ;
64     jTextField24.setText(String.valueOf(df.format(Yt[j])))
65     ;
66     jTextField25.setText(String.valueOf(df.format(Zt[j])))
67     ;
68 }

```

```

53     }
54     jTextField4.setText(String.valueOf(df.format(R11[0])));
55     jTextField5.setText(String.valueOf(df.format(R12[0])));
56     jTextField6.setText(String.valueOf(df.format(R13[0])));
57     jTextField7.setText(String.valueOf(df.format(R14[0])));
58     jTextField18.setText(String.valueOf(df.format(R11[100])));
59     ;
60     jTextField19.setText(String.valueOf(df.format(R12[100])));
61     ;
62     jTextField20.setText(String.valueOf(df.format(R13[100])));
63     ;
64     jTextField21.setText(String.valueOf(df.format(R14[100])));
65     ;
66 }
67 if(Jmove==true){
68     M = (float) Math.sqrt(Math.pow(PlatX/2,2) + Math.pow(
69         PlatY/2,2) + Math.pow(PlatZ/2,2));
70     //comprimento dos cabos inicial
71     R11i = (float) Math.sqrt(Math.pow((Xi-(PlatX/2)-0),2)+
72         Math.pow((Yi-(PlatY/2)-0),2)+ Math.pow((Zi-(PlatZ/2)-
73         EstZ),2));
74     R12i = (float) Math.sqrt(Math.pow((Xi+(PlatX/2)- EstX),2)
75         +Math.pow((Yi-(PlatY/2)- 0),2)+Math.pow((Zi-(PlatZ/2)
76         -EstZ),2));
77     R13i = (float) Math.sqrt(Math.pow((Xi+(PlatX/2)- EstX),2)
78         +Math.pow((Yi+(PlatY/2)- EstY),2)+Math.pow((Zi-(PlatZ
79         /2)-EstZ),2));
80     R14i = (float) Math.sqrt(Math.pow((Xi-(PlatX/2)-0),2)+
81         Math.pow((Yi+(PlatY/2)- EstY),2)+Math.pow((Zi-(PlatZ
82         /2)-EstZ),2));
83     //comprimento dos cabos final
84     R11f = (float) Math.sqrt(Math.pow((Xf-(PlatX/2)-0),2)+
85         Math.pow((Yf-(PlatY/2)-0),2)+ Math.pow((Zf-(PlatZ/2)-
86         EstZ),2));
87     R12f = (float) Math.sqrt(Math.pow((Xf+(PlatX/2)- EstX),2)
88         +Math.pow((Yf-(PlatY/2)- 0),2)+Math.pow((Zf-(PlatZ/2)
89         -EstZ),2));
90     R13f = (float) Math.sqrt(Math.pow((Xf+(PlatX/2)- EstX),2)
91         +Math.pow((Yf+(PlatY/2)- EstY),2)+Math.pow((Zf-(PlatZ
92         /2)-EstZ),2));
93     R14f = (float) Math.sqrt(Math.pow((Xf-(PlatX/2)-0),2)+
94         Math.pow((Yf+(PlatY/2)- EstY),2)+Math.pow((Zf-(PlatZ
95         /2)-EstZ),2));
96     //imprimindo os comprimentos obtidos
97     jTextField4.setText(String.valueOf(df.format(R11i)));
98     jTextField5.setText(String.valueOf(df.format(R12i)));

```

```

79     jTextField6.setText(String.valueOf(df.format(Rl3i)));
80     jTextField7.setText(String.valueOf(df.format(Rl4i)));
81     jTextField18.setText(String.valueOf(df.format(Rl1f)));
82     jTextField19.setText(String.valueOf(df.format(Rl2f)));
83     jTextField20.setText(String.valueOf(df.format(Rl3f)));
84     jTextField21.setText(String.valueOf(df.format(Rl4f)));
85     //calculando os coeficientes do polinomio
86     A01 = Rl1i;
87     A11=0;
88     A21=0;
89     A31=(float) ((20*(Rl1f-Rl1i))/(2*Math.pow(Tf,3)));
90     A41=(float) ((30*(Rl1i-Rl1f))/(2*Math.pow(Tf,4)));
91     A51=(float) ((12*(Rl1f-Rl1i))/(2*Math.pow(Tf,5)));
92
93     A02 = Rl2i;
94     A12=0;
95     A22=0;
96     A32=(float) ((20*(Rl2f-Rl2i))/(2*Math.pow(Tf,3)));
97     A42=(float) ((30*(Rl2i-Rl2f))/(2*Math.pow(Tf,4)));
98     A52=(float) ((12*(Rl2f-Rl2i))/(2*Math.pow(Tf,5)));
99
100    A03 = Rl3i;
101    A13=0;
102    A23=0;
103    A33=(float) ((20*(Rl3f-Rl3i))/(2*Math.pow(Tf,3)));
104    A43=(float) ((30*(Rl3i-Rl3f))/(2*Math.pow(Tf,4)));
105    A53=(float) ((12*(Rl3f-Rl3i))/(2*Math.pow(Tf,5)));
106
107    A04 = Rl4i;
108    A14=0;
109    A24=0;
110    A34=(float) ((20*(Rl4f-Rl4i))/(2*Math.pow(Tf,3)));
111    A44=(float) ((30*(Rl4i-Rl4f))/(2*Math.pow(Tf,4)));
112    A54=(float) ((12*(Rl4f-Rl4i))/(2*Math.pow(Tf,5)));
113
114
115    CommunicationMatlab C = new CommunicationMatlab();
116
117    for ( float i = 0; i<Tf; i=(float) (i+Tf/100)){
118        try {
119
120            Rl1[Indice] = M +(float) ((float) A01+A11*i+A21*
                Math.pow(i,2)+A31*Math.pow(i,3)+A41*Math.pow(
                i,4)+A51*Math.pow(i,5));
121            Rl2[Indice] = M +(float) ((float) A02+A12*i+A22*
                Math.pow(i,2)+A32*Math.pow(i,3)+A42*Math.pow(
                i,4)+A52*Math.pow(i,5));

```

```

122         R13[Indice] = M +(float) ((float) A03+A13*i+A23*
           Math.pow(i,2)+A33*Math.pow(i,3)+A43*Math.pow(
           i,4)+A53*Math.pow(i,5));
123         R14[Indice] = M +(float) ((float) A04+A14*i+A24*
           Math.pow(i,2)+A34*Math.pow(i,3)+A44*Math.pow(
           i,4)+A54*Math.pow(i,5));
124         //System.out.print("\nr1="+R11[Indice]+" \nr2="+
           R12[Indice]+" \nr3="+R13[Indice]+" \nr4="+R14[
           Indice]);
125
126         C.recebe(R11[Indice], R12[Indice], R13[Indice],
           R14[Indice], EstX, EstY, EstZ, Xi, Yi, Zi);
127         Object Resultado = C.envia();
128         Xt[Indice] = (float) ((double[]) Resultado)[0];
129         Yt[Indice] = (float) ((double[]) Resultado)[1];
130         Zt[Indice] = (float) ((double[]) Resultado)[2];
131         jTextField23.setText(String.valueOf(df.format(Xt[
           Indice])));
132         jTextField24.setText(String.valueOf(df.format(Yt[
           Indice])));
133         jTextField25.setText(String.valueOf(df.format(Zt[
           Indice])));
134     } catch (MatlabConnectionException |
           MatlabInvocationException ex) {
135         Logger.getLogger(ControladorEditor.class.getName
           ()).log(Level.SEVERE, null, ex);
136     }
137     Indice++;
138 }
139
140 }

```

## C.2 Código da Classe que Realiza o Desenho da Janela de Simulação do Sistema

```

1 public class DrawLines extends JPanel implements Runnable {
2
3     int X[]= new int[105];
4     int Y[]= new int[105];
5     int Z[]= new int[105];
6     int Xa, Ya, Za, Xp, Yp, Zp;
7     int Platx,PlatY,Platz,EstX,EstY,EstZ;
8     int NEX, NEY, NEZ, NPX, NPY, NPZ, ZZ, Xf, Tf, Tfi;
9     boolean aberta, fechada;
10    Graphics g;
11    Thread thread = new Thread(this);

```



```

12
13 public void trajetoria(float[] Xx, float[] Yy, float[] Zzz,
    int Platxx, int PlatYy, int PlatZz, int EstXx, int EstYy,
    int EstZz, float T, boolean abertan, boolean fechadan){
14     Platx= Platxx;
15     PlatY= PlatYy;
16     PlatZ= PlatZz;
17     EstX = EstXx;
18     EstY = EstYy;
19     EstZ = EstZz;
20     aberta = abertan;
21     fechada = fechadan;
22     Tf = (int) T;
23     Tfi = ((Tf*1000)/101)+10;
24     for(int i=0; i<105; i++){
25         X[i]= (int) Xx[i];
26         Y[i]= (int) ((int) EstY-Yy[i]);
27         Z[i]= (int) ((int) EstZ-Zzz[i]);
28         if(Z[i]<PlatZ/2){
29             Z[i]=PlatZ/2;
30         }
31         if(Z[i]>EstZ-PlatZ/2){
32             Z[i]=EstZ-PlatZ/2;
33         }
34         if(X[i]<Platx/2){
35             X[i]=Platx/2;
36         }
37         if(X[i]>EstX-Platx/2){
38             X[i]=EstX-Platx/2;
39         }
40         if(Y[i]<PlatY/2){
41             Y[i]=PlatY/2;
42         }
43         if(Y[i]>EstY-PlatY/2){
44             Y[i]=EstY-PlatY/2;
45         }
46     }
47
48     thread.start();
49
50 }
51
52 @Override
53 public void run() {
54     for(int i=0; i<101; i++){
55         Xa = X[i];
56         Xp = X[i+1];

```

```

57         Ya = Y[i];
58         Yp = Y[i+1];
59         Za = Z[i];
60         Zp = Z[i+1];
61         repaint();
62         try {
63             Thread.sleep(Tfi);
64         } catch (InterruptedException ex) {
65             Logger.getLogger(DrawLines.class.getName()).log(
66                 Level.SEVERE, null, ex);
67         }
68     }
69
70     public void paintComponent( Graphics g){
71
72         super.paintComponent( g );
73         //trajetoria
74         g.setColor(Color.RED);
75         g.drawLine(X[1]+20, Y[1]+20, X[5]+20, Y[5]+20);
76         g.drawLine(X[5]+20, Y[5]+20, X[10]+20, Y[10]+20);
77         g.drawLine(X[10]+20, Y[10]+20, X[15]+20, Y[15]+20);
78         g.drawLine(X[15]+20, Y[15]+20, X[20]+20, Y[20]+20);
79         g.drawLine(X[20]+20, Y[20]+20, X[25]+20, Y[25]+20);
80         g.drawLine(X[25]+20, Y[25]+20, X[30]+20, Y[30]+20);
81         g.drawLine(X[30]+20, Y[30]+20, X[35]+20, Y[35]+20);
82         g.drawLine(X[35]+20, Y[35]+20, X[40]+20, Y[40]+20);
83         g.drawLine(X[40]+20, Y[40]+20, X[45]+20, Y[45]+20);
84         g.drawLine(X[45]+20, Y[45]+20, X[50]+20, Y[50]+20);
85         g.drawLine(X[50]+20, Y[50]+20, X[55]+20, Y[55]+20);
86         g.drawLine(X[55]+20, Y[55]+20, X[60]+20, Y[60]+20);
87         g.drawLine(X[60]+20, Y[60]+20, X[65]+20, Y[65]+20);
88         g.drawLine(X[65]+20, Y[65]+20, X[70]+20, Y[70]+20);
89         g.drawLine(X[70]+20, Y[70]+20, X[75]+20, Y[75]+20);
90         g.drawLine(X[75]+20, Y[75]+20, X[80]+20, Y[80]+20);
91         g.drawLine(X[80]+20, Y[80]+20, X[85]+20, Y[85]+20);
92         g.drawLine(X[85]+20, Y[85]+20, X[90]+20, Y[90]+20);
93         g.drawLine(X[90]+20, Y[90]+20, X[95]+20, Y[95]+20);
94         g.drawLine(X[95]+20, Y[95]+20, X[100]+20, Y[100]+20);
95         g.setColor(Color.BLACK);
96         g.drawString("Posicao atual da garra:", (((EstX+EstX+80)
97             /2)-130), 15);
98         g.drawString("X="+Xa+" -- Y="+Math.abs(Ya-EstY)+" -- Z="+
99             Math.abs(Za-EstZ), ((EstX+EstX+80)/2), 15 );
100        //x-y
101        g.setColor(Color.lightGray);
102        g.drawRect(20, 20, EstX, EstY); //estrutura

```

```

101 g.setColor(Color.DARK_GRAY);
102 g.drawLine(20,20,(Xa+20 -(Platx/2)),Ya+20-(PlatY/2)); //a4
103 g.drawLine(20,EstY+20,(Xa+20-(Platx/2)),(Ya+20+(PlatY/2))
    ); //a1
104 g.drawLine(EstX+20,20,(Xa+20+(Platx/2)),(Ya+20-(PlatY/2))
    ); //a3
105 g.drawLine(EstX+20,EstY+20,(Xa+20+(Platx/2)),(Ya+20+(
    PlatY/2))); //a2
106 g.setColor(Color.BLACK);
107 g.drawRect( (Xa+20-(Platx/2)),(Ya+20-(PlatY/2)),Platx,
    PlatY); //plataforma
108 g.setColor(Color.GREEN);
109 g.drawLine(Xa+20, Ya+20, Xa+20, Ya-5); //tcpY
110 g.setColor(Color.BLUE);
111 g.drawLine(Xa+20, Ya+20, Xa+45, Ya+20); //tcpX
112
113 //x-z
114 //trajetoria
115 g.setColor(Color.RED);
116 g.drawLine(X[1]+40+EstX, Z[1]+20, X[5]+40+EstX, Z[5]+20);
117 g.drawLine(X[5]+40+EstX, Z[5]+20, X[10]+40+EstX, Z
    [10]+20);
118 g.drawLine(X[10]+40+EstX, Z[10]+20, X[15]+40+EstX, Z
    [15]+20);
119 g.drawLine(X[15]+40+EstX, Z[15]+20, X[20]+40+EstX, Z
    [20]+20);
120 g.drawLine(X[20]+40+EstX, Z[20]+20, X[25]+40+EstX, Z
    [25]+20);
121 g.drawLine(X[25]+40+EstX, Z[25]+20, X[30]+40+EstX, Z
    [30]+20);
122 g.drawLine(X[30]+40+EstX, Z[30]+20, X[35]+40+EstX, Z
    [35]+20);
123 g.drawLine(X[35]+40+EstX, Z[35]+20, X[40]+40+EstX, Z
    [40]+20);
124 g.drawLine(X[40]+40+EstX, Z[40]+20, X[45]+40+EstX, Z
    [45]+20);
125 g.drawLine(X[45]+40+EstX, Z[45]+20, X[50]+40+EstX, Z
    [50]+20);
126 g.drawLine(X[50]+40+EstX, Z[50]+20, X[55]+40+EstX, Z
    [55]+20);
127 g.drawLine(X[55]+40+EstX, Z[55]+20, X[60]+40+EstX, Z
    [60]+20);
128 g.drawLine(X[60]+40+EstX, Z[60]+20, X[65]+40+EstX, Z
    [65]+20);
129 g.drawLine(X[65]+40+EstX, Z[65]+20, X[70]+40+EstX, Z
    [70]+20);
130 g.drawLine(X[70]+40+EstX, Z[70]+20, X[75]+40+EstX, Z

```

```

    [75]+20);
131 g.drawLine(X[75]+40+EstX, Z[75]+20, X[80]+40+EstX, Z
    [80]+20);
132 g.drawLine(X[80]+40+EstX, Z[80]+20, X[85]+40+EstX, Z
    [85]+20);
133 g.drawLine(X[85]+40+EstX, Z[85]+20, X[90]+40+EstX, Z
    [90]+20);
134 g.drawLine(X[90]+40+EstX, Z[90]+20, X[95]+40+EstX, Z
    [95]+20);
135 g.drawLine(X[95]+40+EstX, Z[95]+20, X[100]+40+EstX, Z
    [100]+20);
136 //end trajetoria
137 g.setColor(Color.lightGray);
138 g.drawRect(40+EstX, 20, EstX, EstZ); //estrutura
139 g.setColor(Color.BLACK);
140 g.drawRect((Xa+40+EstX-(Platx/2)), Za+20-(Platz/2), Platx,
    Platz/4); //plataforma
141 g.setColor(Color.DARK_GRAY);
142 g.drawLine(40+EstX, 20, (Xa+40+EstX-(Platx/2)), Za+20-(
    Platz/2)); //cabos
143 g.drawLine(40+EstX+EstX, 20, (Xa+40+EstX+(Platx/2)), Za
    +20-(Platz/2)); //cabos
144 g.drawLine(Xa+40+EstX, Za+20-Platz/4, Xa+40+EstX, Za+20-
    Platz/8); //mancal
145 g.drawLine(Xa+40+EstX, Za+20-Platz/8, Xa+40+EstX+Platx/2,
    Za+20-Platz/8);
146 g.drawLine(Xa+40+EstX, Za+20-Platz/8, Xa+40+EstX-Platx/2,
    Za+20-Platz/8);
147 g.setColor(Color.BLACK);
148 if(aberta == true){
149 g.drawLine(Xa+40+EstX+Platx/2, Za+20-Platz/8, Xa+40+EstX+
    Platx/2, Za+20+Platz/2);
150 g.drawLine(Xa+40+EstX-Platx/2, Za+20-Platz/8, Xa+40+EstX-
    Platx/2, Za+20+Platz/2);
151 }
152 if(fechada == true){
153 g.drawLine(Xa+40+EstX+Platx/2, Za+20-Platz/8, Xa+40+EstX+
    Platx/4, Za+20+Platz/2);
154 g.drawLine(Xa+40+EstX-Platx/2, Za+20-Platz/8, Xa+40+EstX-
    Platx/4, Za+20+Platz/2);
155 }
156 g.setColor(Color.YELLOW);
157 g.drawLine(Xa+40+EstX, Za+20, Xa+40+EstX, Za-5); //tcpZ
158 g.setColor(Color.BLUE);
159 g.drawLine(Xa+40+EstX, Za+20, Xa+65+EstX, Za+20); //tcpX
160 g.setColor(Color.BLACK);
161 g.drawLine(10, EstY+30, 10, EstY);

```

```

162     g.drawLine(10,EstY+30, 40, EstY+30);
163     g.drawLine(10, EstY, 15, EstY+5);
164     g.drawLine(10, EstY, 5, EstY+5);
165     g.drawLine(40, EstY+30, 35, EstY+35);
166     g.drawLine(40, EstY+30, 35, EstY+25);
167     g.drawLine(EstX+30,EstZ+30,EstX+30, EstZ);
168     g.drawLine(EstX+30,EstZ+30, EstX+60, EstZ+30);
169     g.drawLine(EstX+30, EstZ, EstX+25, EstZ+5);
170     g.drawLine(EstX+30, EstZ, EstX+35, EstZ+5);
171     g.drawLine(EstX+60, EstZ+30, EstX+55, EstZ+35);
172     g.drawLine(EstX+60, EstZ+30, EstX+55, EstZ+25);
173     g.setColor(Color.BLACK);
174     g.drawString("Z",EstX+28 , EstZ-2);
175     g.drawString("X",EstX+65 ,EstZ+35);
176     g.drawString("Y",8 , EstY-5);
177     g.drawString("X",45 ,EstY+35);
178 }

```

### C.3 Código da Classe de Comunicação com o Matlab®

```

1  public class CommunicationMatlab {
2
3      String Rr1;
4      String Rr2;
5      String Rr3;
6      String Rr4;
7      String EestX;
8      String EestY;
9      String EestZ;
10     String Xxi, Yyi, Zzi;
11
12     public void recebe(float R1, float R2, float R3, float R4,
13         int EstX, int EstY, int EstZ, int Xi, int Yi, int Zi){
14         Rr1 = Float.toString(R1);
15         Rr2 = Float.toString(R2);
16         Rr3 = Float.toString(R3);
17         Rr4 = Float.toString(R4);
18         EestX = Integer.toString(EstX);
19         EestY = Integer.toString(EstY);
20         EestZ = Integer.toString(EstZ);
21         Xxi = Integer.toString(Xi);
22         Yyi = Integer.toString(Yi);
23         Zzi = Integer.toString(Zi);
24     }
25
26     public Object envia () throws MatlabConnectionException,
27         MatlabInvocationException {

```

```

26     MatlabProxyFactoryOptions options = new
        MatlabProxyFactoryOptions.Builder()
27         .setUsePreviouslyControlledSession(true)
28         .setHidden(true)
29         .setMatlabLocation(null).build();
30     MatlabProxyFactory factory = new MatlabProxyFactory(
        options);
31     MatlabProxy proxy = factory.getProxy();
32     String f = "fsolve(@(x) [x(1)^2+x(2)^2+(x(3)-"+EestZ+"")
        ^2-("+Rr1+")^2; "
33         + "(x(1)-"+EestX+")^2+x(2)^2+(x(3)-"+EestZ+")^2-("
        "+Rr2+")^2; "
34         + "(x(1)-"+EestX+")^2+(x(2)-"+EestY+")^2+(x(3)-"+
        EestZ+")^2-("+Rr3+")^2; "
35         + "x(1)^2+(x(2)-"+EestY+")^2+(x(3)-"+EestZ+")^2-("
        "+Rr4+")^2] , "
36         + "["+Xxi+"; "+Yyi+"; "+Zzi+"])";
37
38     Object[] d = proxy.returningEval(f,1);
39     Object arg = d[0];
40     double x = ((double[]) arg)[0];
41     double y = ((double[]) arg)[1];
42     double z = ((double[]) arg)[2];
43     proxy.disconnect();
44     return arg;
45 }
46 }

```

